UNIVERSIDAD CARLOS III DE MADRID

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA
DEPARTAMENTO DE AUTOMÁTICA Y ROBÓTICA

TRABAJO FIN DE GRADO

STRUCTURE FROM MOTION AND 3D WORLD RECONSTRUCTION USING UAVS

AUTOR: ÁLVARO APONTE CABRERA
TUTOR: ABDULLA HUSSEIN AL-KAFF

Octubre de 2015

Título: *STRUCTURE FROM MOTION AND 3D WORLD RECONSTRUCTION USING UAVS.*

Autor: *ÁLVARO APONTE CABRERA*

Tutor: *ABDULLA HUSSEIN AL-KAFF*

La defensa del presente Proyecto Fin de Carrera se realizó el día 14 de Octubre de 2015; siendo calificada por el siguiente tribunal:

Presidente:

Secretario

Vocal

Habiendo obtenido la siguiente calificación:

Calificación:

Presidente **Secretario** **Vocal**

Agradecimientos

Quiero dedicar este proyecto a mi familia y a las personas importantes que tengo a mi lado, sin las cuales no habría conseguido nada de lo que tengo.

También quiero dedicar este proyecto a los profesores que he tenido a lo largo de mi carrera y en especial a mi tutor Abdulla Hussein Al-Kaff por su apoyo y paciencia.

Muchas gracias a todos, sin vosotros ésto no habría sido posible.

Nuestras virtudes y nuestros defectos son inseparables, como la fuerza y la materia.
Cuando se separan, el Hombre no existe

Nikola Tesla

Resumen

El objetivo de este trabajo es la obtención de una reconstrucción 3D del ambiente. Para ello se construye un algoritmo basado en el concepto Structure from Motion para el estudio de imágenes en movimiento.

En primer lugar, se van a explicar los conceptos de UAV y los métodos de obtención de keypoints y descriptores, la comparación entre ellos y el filtrado de keypoints no comunes. También se va a explicar el concepto de reconstrucción 3D y los pasos intermedios necesarios como son los conceptos de matriz fundamental y esencial, matrices de rotación y translación y triangulación.

Por otro lado se expondrá un algoritmo de detección de los puntos reales, explicando en cada etapa las entradas, salidas y objetivos de forma detallada. Este algoritmo podrá ser aplicado en cualquier plataforma de programación ya que aquí sólo se exponen los fundamentos teóricos.

Para terminar, se decidirá qué método de extracción de keypoints y descriptores se ha elegido, justificando la elección con ejemplos. También se va a mostrar ejemplos de la reconstrucción 3D explicando los resultados obtenidos.

Como información complementaria se va a explicar el algoritmo utilizado para la calibración de la cámara.

Abstract

The purpose of this project is to produce a 3D reconstruction of the environment. For this, an algorithm based on Structure from Motion is built to study moving images.

First, is to explain the concepts of UAV and methods of obtaining keypoints and descriptors, the comparison between them and the filtering keypoints uncommon. It will also explain the concept of 3D reconstruction and intermediate concepts fundamental and essential matrix, rotation and translation matrices and triangulation.

On the other hand a detection algorithm in real sections describe. Each step is explained the inputs, outputs and objectives in detail. This algorithm can be implemented in any programming language because here only the theoretical foundations exposed.

Finally, it will decide which method of extraction of keypoints and descriptors was chosen, justifying the choice examples. It will also show examples of 3D reconstruction explaining the results.

Within the appendices you were going to explain how to obtain matrix camera calibration. Another section will also present the budget.

Índice general

Lista de Figuras

Lista de Tablas

Lista de algoritmos

Capítulo 1

INTRODUCCIÓN

Este proyecto ha sido desarrollado para investigar el uso del algoritmo de tratamiento de imágenes en tiempo real denominado Structure from Motion, en concreto en el uso de la reconstrucción 3D de los puntos característicos del ambiente. Mediante la reconstrucción 3D se podrá obtener la representación de los puntos reales característicos del ambiente.

1.1. MOTIVACIONES

En el mundo tecnológico en el que vive el ser humano los desarrollos en este campo se han convertido en la forma de avanzar en el estado del bienestar. La vida de hace tres décadas hasta hoy ha cambiado de manera importante, desde la aparición de los primeros móviles hasta los smartphones actuales. Esta evoluación constante ha convertido al ser humano en un devorador de la nueva tecnología y de los últimos avances.

Desde hace un tiempo a esta parte han aparecido en el mercado un nuevo competidor, los UAV o más comúnmente conocidos como drones. Esta tecnología une muchas ramas de investigación actuales como son el campo de la robótica, la comunicación o de la tecnología de aviación, entre otros. Está creciendo cada vez más el uso de los drones civiles para uso lúdico en un mercado que no para de ofrecer nuevos usos para estas máquinas. Pero esta evolución conlleva un avance en el ámbito comercial. Los drones se han visto como una buena herramienta para el trabajo.

Los drones son máquinas que pueden ser dirigidas por control remoto o navegar de

forma autónoma, pueden portar herramientas específicas para trabajos concretos, puede acceder a sitios donde el hombre no llega, reduce costes ante alternativas tan caras como helicópteros o avionetas, etc. Debido a esta versatilidad muchos programadores están centrando sus estudios en el desarrollo de usos y aplicaciones de estos aparatos. En cierta forma se puede hacer una comparación con la aparición de los primeros smartphones y su forma de cambiar la vida de las personas medias. El mercado está todavía en fase de construcción y muchas personas ven en estas máquinas una gran posibilidad de explotación, tanto económica como tecnológica.

Uno de los campos que más futuro tiene dentro de los drones es el análisis de imágenes a tiempo real. Esta tecnología ofrece algo que hasta entonces no existía: se puede llevar una cámara a casi cualquier zona con un coste tanto económico como en personal relativamente bajo. Debido a esto muchos de los avances para los drones tienen que ver en el análisis y tratamiento de imágenes para muchas situaciones como pueden ser actividades deportivas, análisis de terrenos o cartografía, por poner algún ejemplo. La motivación de este proyecto es debida en parte a esta situación, la aportación de nuevas herramientas para el uso de estas máquinas. El objetivo es la obtención de un algoritmo que utilice la cámara para la obtención de las características del ambiente.

Para ello se hará uso de algoritmos de tratamiento de imágenes en movimiento como Structure from Motion y reconstrucción 3D del ambiente. También se pretende aportar todo el fundamento teórico que este estudio conlleva y la creación de una estructura aplicable a cualquier lenguaje de programación.

1.2. MARCO DEL PROYECTO

Este proyecto ha sido desarrollado en el departamento de Ingeniería de Sistemas y Automática de la Escuela Politécnica Superior de la Universidad Carlos III de Madrid como trabajo de fin de grado, correspondiente al Grado en Ingeniería Electrónica Industrial y Automática.

1.3. OBJETIVOS

El objetivo del proyecto es la obtención de los puntos característicos del ambiente mediante una reconstrucción 3D. Para conseguirlo, se deben atender a las siguientes etapas:

- Obtención de los puntos característicos. Se obtienen los puntos representativos de cada imagen, así como sus características. Se van a exponer varios métodos de extracción de características de los puntos y se elegirá la más adecuada.

- Comparación de los puntos y obtención de los valores característicos. Una vez se tienen los puntos característicos de las imágenes, se procede a compararlos en cada imagen. Mediante este método se obtienen aquellos que pertenecen a las dos imágenes y se eliminan el resto. Con estos puntos puede obtenerse la matriz fundamental.

- Obtención de la matriz de calibración de la cámara y la matriz esencial. Se obtiene la matriz de calibración de la cámara y se opera con la matriz fundamental, obteniendo así los valores de la posición y de la rotación de los puntos.

- Obtención de los puntos reales. Con la matriz esencial, se pueden hallar mediante los puntos y líneas epipolares las posiciones reales de los puntos.

Dentro de las secciones del estado del arte y el desarrollo teórico se profundizan en todos estos elementos así como en otros necesarios para la explicación. En el apartado de resultados se justificará con ejemplos el algoritmo usado. También se expondrán las conclusiones y las futuras líneas de estudio dentro de la materia.

1.4. ESTRUCTURA DE LA MEMORIA

La memoria está estructurada en cinco grandes bloques. Se va a exponer un breve resumen de cada uno a continuación:

- Capítulo 1 Introducción. Este capítulo introduce las motivaciones para la elaboración del proyecto y los objetivos que se pretenden alcanzar.

- Capítulo 2 Estado del arte. Este capítulo muestra la base teórica necesaria para la comprensión y elaboración del proyecto. Se define el concepto de Structure from Motion en el que se basa el algoritmo. se define el concepto de UAV y su clasificación, así como su funcionamiento. También se define el concepto de punto de interés, para quéSSS sirve y varios métodos de obtención como son los métodos SIFT, SURF, FAST y FREAK.

- Capítulo 3 Extracción del algoritmo. En este capítulo se define el algoritmo de obtención de los puntos reales y su muestreo. Para ello se expone cada una de las partes en las que está dividido, como son la extracción de los keypoints, la obtención de la matriz fundamental y esencial, las matrices de rotación y traslación y los puntos reales.

- Capítulo 4 Resultados prácticos. En este capítulo se introducen las pruebas prácticas que se obtiene de la aplicación del algoritmo. Se va a justificar la elección del método de extracción de keypoints y también mostrar los resultados de la reconstrucción de los puntos reales.

- Capítulo 5 Conclusiones y trabajos futuros. Se exponen las conclusiones del proyecto y también se exponen las líneas de investigación futuras por las que puede avanzar el proyecto.

Además se exponen dos anexos, el primero define la forma de calibrar una cámara y la segunda expone el presupuesto del proyecto.

Capítulo 2

ESTADO DEL ARTE

2.1. INTRODUCCIÓN A LOS UAVs

Los UAV, siglas de *Unmanned Aerial Vehicle*, en castellano VANT, *Vehículo Aéreo No Tripulado*, se define según la agencia de defensa del ejército de los Estados Unidos [2] como:

> Un vehículo aéreo que no lleva a bordo a un operador humano, utiliza las fuerzas aerodinámicas para generar la sustentación, puede volar de forma autónoma o remota, que puede ser fungible o recuperable, y que puede transportar una carga de pago letal o no.

Esta definición excluye como UAV a los misiles balísticos, misiles de tipo crucero y proyectiles de artillería. De igual forma se excluyen los planeadores (no llevan planta propulsora) o los globos y dirigibles (no usan formas de propulsión mediante fuerzas aerodinámicas sino de flotabilidad).

2.1.1. CONTEXTO HISTÓRICO

A lo largo de la historia se ha intentado conseguir un arma capaz de generar daños sin arriesgar ninguna vida humana, ni de pilotos ni de tripulación. Es a partir de la la Segunda Guerra Mundial y con el paso de los años que los ejércitos de las dos superpotencias de

la época (Estados Unidos y la Unión Soviética) investigan en nuevas naves que puedan ser pilotadas desde tierra. De esta forma se consigue acabar con objetivos enemigos sin necesidad de arriesgar la vida de ningún soldado[19].

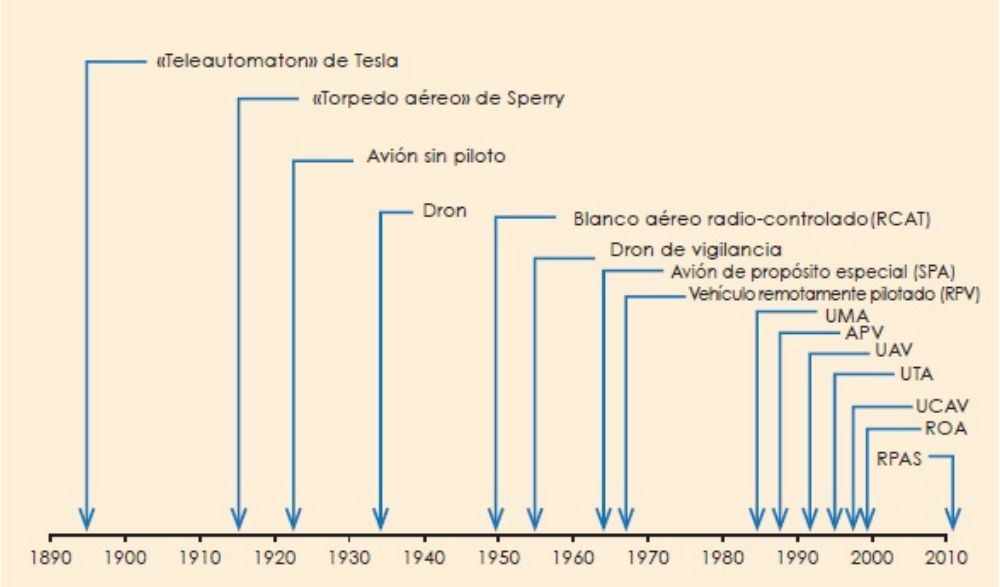


Figura 2.1: *Cronología de la evolución de las aeronaves de vuelo autónomo.*

Naves desarrolladas durante la Guerra Fría.

Durante la Segunda Guerra Mundial y en las décadas siguientes se experimenta con la obtención de un UAV: una nave capaz de ser controlada desde tierra. Los primeros modelos fueron avionetas controladas mediante radiofrecuencia. El problema de estos sistemas era su fiabilidad y su corto alcance, siendo usados en forma de experimentación.

Con el paso de los años y la invención de los aviones a reacción se vuelve a intensificar el estudio del manejo remoto de estos aviones. Durante la década de los sesenta se introducen dispositivos de captura de vídeos, usándose para misiones de reconocimiento y como aviones espía. Este uso resultó muy útil ya que impedía problemas diplomáticos al no poder ser capturado ningún piloto.

Estas naves tenían varios inconvenientes: era muy difícil aterrizar de forma que se pudieran recuperar, por lo que no eran reutilizables, causando grandes pérdidas de dinero.

Otro de los problemas era la limitación de la tecnología del momento. El nivel de innovación era bastante limitado, por lo que este sistema fue cayendo en el desuso gradualmente.

Fue en la década de los años 80 y con la mejora de la tecnología cuando se intensificó el uso de los UAV. Se desarrollaron sistemas de vuelo pre-programados para las naves y se empezaron a utilizar naves impulsadas por rotores (helicópteros). De esta manera, se fueron introduciendo cada vez más sensores que hacían más fácil la navegación y el control desde tierra.

Estos elementos fueron utilizados para labores de espionaje y también para ataques electrónicos e informáticos. De forma paralela, se buscaba una mejora en la precisión y tiempo de acción de los UAV militares. Si bien su sistema de vuelo se podía calificar como aceptable mediante el de uso sensores modernos(cámaras en tiempo real y primeros y rudimentarios GPS), la selección de blancos no era la óptima.

Llegando la Guerra Fría a sus últimos años en la década de los años 80, los ataques espía y de reconocimiento eran una de las mayores armas para los dos bandos, por lo que una vez finalizada esta época se apostó por seguir con la investigación.

Década de los años 90.

Una vez la Guerra Fría llega a su fin, el estudio de naves sin tripulación sigue aumentando por parte del ejército estadounidense. Se introducen varias mejoras sobre las naves: uso del GPS de manera óptima, uso de los sistemas digitales de control de vuelo (DFCS), vuelo a mayor altura evitando que la nave sea descubierta.

En esta época nace el que se considera como el primer UAV militar funcional denominado *MALE UAS Predator*. Se trata de una aeronave de gran envergadura, propulsión por reactor y capacidad de ser controlada desde tierra a tiempo real. Su utilidad radica en el vuelo a grandes alturas, evitando su detección. También entran en escena las primeras aeronaves de despegue y aterrizaje vertical en Japón con el modelo *Yamaha R50* y el *Yamaha RTOL*. Se tratan de vehículos de pala rotativa de navegación controlada desde tierra, usándose no solo para misiones militares sino también introduciéndose poco a poco en el ámbito civil.

Figura 2.2: *Modelo MQ-Predator A*

Figura 2.3: *Modelo Yamaha R50*

Años 2000. Expansión de los UAV.

A partir de la salida de los modelos militares en la década de los años 90 se empiezan a mejorar y a usar cada vez más en el campo de batalla, siendo una parte importante su utilización en la eliminación selectiva de enemigos y en la preparación del terreno para una incursión terrestre. Se hacen mejoras en las aeronaves como la instalación de armamento automático y detectores de infrarrojos que facilitan esta tarea.

De forma paralela a este desarrollo, se empieza a ver la utilidad de los UAV en la vida civil. El uso de las grandes aeronaves en el ámbito civil se ve frenado no por un problema técnico, sino por uno moral: es difícil distinguir entre nave tripulada y no tripulada, dando pie a debates éticos sobre su uso indebido. Algunas empresas empiezan a desembolsar grandes cantidades de dinero para su investigación y aplicación. Un ejemplo de estas

empresas es la NASA, que se interesa por el uso de estas naves para el estudio de las capas altas de la atmósfera, por lo que intenta adquirir naves para este uso.

Por otro lado, se intentan potenciar el uso de las naves de despegue y aterrizaje vertical debido a su mejor uso y menor índice de accidentes en comparación con los UAV de despegue vertical, en especial en los aterrizajes y despegues.

Año 2010 y futuro. UAV comerciales.

Pasada la primera década del milenio, aparece en escena un nuevo elemento: Los UAV de uso comercial. En especial, aparecen unos nuevos UAV que por tamaño y manejo causan un gran impacto entre el público, los denominados *drones*. Estos aparatos, de pequeño tamaño y fácil uso mediante radio control, tienen una gran aceptación en el uso de actividades deportivas u obtención de imágenes.

Su utilidad no queda ahí, ya que las posibilidades de uso de estos elementos son incontables, tanto para un uso lúdico como para uso profesional. Nace una nueva vertiente de investigadores que se dedica a desarrollar estos elementos para su uso en las empresas y cada año se multiplican sus ventas.

En la actualidad la investigación del desarrollo de drones por parte de la empresa privada y las universidades abre un gran marco de negocio, comparable con la aparición de los primeros smartphones. Su desembolso inicial no es muy grande y disponen de una gran versatilidad al poder ser implementados controles mediante plataformas portátiles, como pueden ser las tablets y los smartphones.

En definitiva, el mundo de los drones actualmente cuenta con unas capacidades de evolución enormes y es por ésto que muchos investigadores deciden invertir en esta nueva tecnología. El uso cada vez más intensivo del automatismo en el sector producción hace de esta tecnología una de las grandes bazas de futuro.

Figura 2.4: *Drone multi-rotor de uso civil*

2.1.2. CLASIFICACIÓN

Los UAV pueden ser clasificados según diversos criterios, como pueden ser su forma de despegue, la altitud que alcanzan, la carga máxima o el nivel de autonomía alcanzable[9].

Respecto a este punto, cabe puntualizar la diferencia existente entre el término UAV y el término UAS:

- UAV(Unmanned Aerial Vehicle), abarca a la nave y sus sensores, define al sistema de propulsión, el procesador, la estructura y los sensores. En otras palabras, el término UAV define al dispositivo.

- UAS(Unmanned Aerial System), abarca tanto a la nave y a sus sensores como al sistema de control, que puede ser autónomo o controlado desde tierra. En otras palabras, el UAS abarca el propio UAV y la forma de controlarlo.

Generalmente se suele utilizar el término UAV para designar los sistemas UAS debido a la imposibilidad de distinguirlo a primera vista. En adelante, en este documento se hará referencia al término UAV sin ninguna distinción para simplificar el tema.

En este documento se van a clasificar según su forma de despegue, puesto que dará una mejor aproximación de su aspecto. En la figura 2.5 se definen dos familias de UAV, los que tienen un despegue vertical y los que tienen un despegue no vertical.

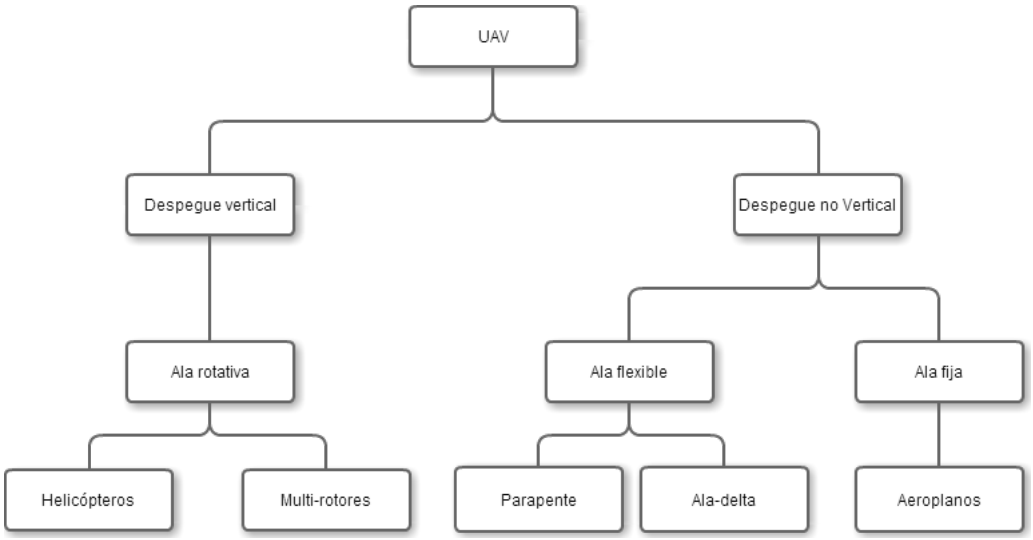


Figura 2.5: *Clasificación de los UAV según su forma de despegue*

Los UAV de despegue vertical pueden definirse prácticamente en su totalidad como máquinas con propulsión por pala rotativa. Dentro de este tipo, los UAV se pueden diferenciar entre helicópteros, cuya estabilidad viene dada por el rotor de cola, y los multi-rotores, cuya estabilidad viene dada por la alternancia de la aceleración de sus rotores.

Los UAV de despegue no vertical pueden clasificarse según la estructura de su ala. Así los aeroplanos, de pala fija, utilizan un propulsor y estabilizan su vuelo con un timonel de cola como hacen los aviones covencionales. Los UAV de ala flexible utilizan un propulsor para desplazarse y pueden estabilizarse mediante un sistema basado en ala delta o en forma de parapente. Los UAV basados en ala delta consiguen su estabilidad variando la superficie de contacto del aire, mientras que aquellos que se basan en parapentes lo hacen controlando la posición del paracaídas.

Los UAV más usado en su uso comercial son los de pala rotativa ya que su despegue y su aterrizaje es más sencillo y menos peligroso. Dentro de los UAV de pala rotativa son más comunes de encontrar los multi-rotores, que ofrecen una buena respuesta a cambios de viento y una mayor estabilidad en estático que los helicópteros.

(a) UAV aeroplano

(b) UAV Ala delta

(c) UAV parapente

(d) UAV helicóptero

(e) UAV multi-rotor

Figura 2.6: *Tipos de UAV según su forma de despegue. Las figuras (a), (b) y (c) corresponden a UAVs de despegue no vertical y las figuras (d) y (e) corresponden a UAVs de despegue vertical*

Estructura de un UAV multi-rotor

Este tipo de UAV tienen una gran aceptación en el mercado, siendo el tipo de UAV más vendido por su sencillez de manejo. Denominado *drone* en inglés y *dron* en castellano, las formas en las que se pueden encontrar son muy variadas, aunque todas tienen una serie de elementos en común.

- Chasis. Es el esqueleto del UAV y donde van montados todos los elementos. Tiene que ser resistente y liviano. Suelen estar fabricados en plástico, fibra de carbono o aluminio, según la resistencia necesaria.

- Controlador de vuelo. Es el cerebro del UAV, contiene una CPU y la electrónica necesarios para su funcionamiento y control. Dentro de este elemento se encuentra el microprocesador encargado del funcionamiento del dispositivo y también otros elementos como acelerómetros, GPS o cualquier otro tipo de sensor necesario para su funcionamiento.

- Motores. Dan propulsión al UAV y también se usan para la estabilización y los giros. El microprocesador regula la velocidad con ayuda de un sistema de control electrónico de la velocidad (ESC) para cada motor. Dependiendo del número de rotores que disponga en UAV se denominan de formas diferentes: cuadricópteros (4 rotores), hexacópteros (6 rotores) u octacópteros (8 rotores). Los motores se compensan entre sí para ganar estabilidad y para realizar giros e inclinaciones.

- Batería. Suministra la energía necesaria para alimentar tanto a los motores como al controlador de vuelo. Cuanto mayor sea la capacidad de carga de la batería más pesada será, por lo que la relación entre tiempo de autonomía de vuelo y peso es vital para el diseño del UAV. Las baterías Li-Po suelen ser las más utilizadas por su buena relación entre carga y peso.

- Sistema de control. La mayoría de los UAV en el mercado disponen de un sistema de control desde tierra. Pueden ser controlados mediante un mando receptor de radio o desde una estación de control, generalmente un ordenador o una tablet.

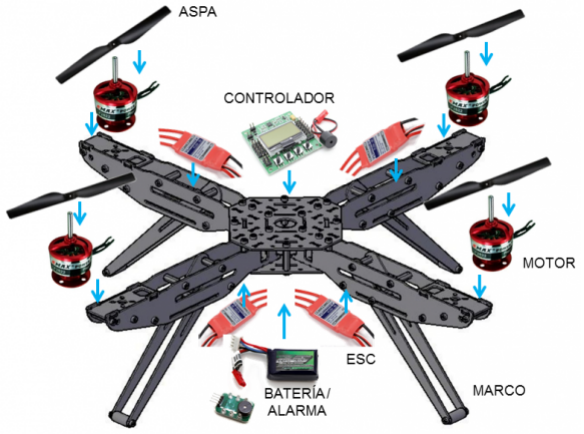


Figura 2.7: *Partes de un UAV*

Todos los UAV disponen de estos elementos para poder funcionar. Adicionalmente se puede introducir accesorios dependiendo del tipo de uso que se le quiera dar.

El accesorio más común es el uso de cámaras. Pueden estar integradas en el Controlador de vuelo o acopladas mediante soportes en el chasis. De la misma forma se pueden acoplar diversos sensores en función del uso que se le quiera dar como cámaras infrarrojas, detectores de calor, etc.

(a) Cámara interna

(b) Cámara adicional

Figura 2.8: *Tipos de cámaras que se pueden encontrar en un UAV. En el caso (a) se encuentra la cámara integrada en el chasis. En el caso (b) la cámara se encuentra acoplada al UAV mediante un soporte añadido.*

2.1.3. APLICACIONES DEL USO DE LA CÁMARA EN UAVs

Una de las mayores utilidades de los UAV es la posibilidad de sobrevolar casi cualquier zona con varias ventajas respecto a las naves pilotadas:

- Pueden acceder a zonas de difícil acceso para naves más grandes.
- Se reduce el costo, tanto de alquiler como de mantenimiento.
- Se minimizan los riesgos para los trabajadores al prescindir de pilotos.

En la mayoría de las aplicaciones desarrolladas entra en escena una cámara integrada en el UAV. El uso de esta cámara es vital, puesto que son los ojos de los operarios. Es por este motivo que una de las partes más importantes del desarrollo de UAV venga por la integración de la captura de imágenes y su análisis.

A modo de información, se van a explicar a continuación algunas aplicaciones de proyectos con UAV donde el uso de tratamientos de imágenes cobra mucha importancia.

Aplicación de UAVs en la agricultura

El uso de tecnología para mejorar la producción es un tema recurrente a lo largo de los últimos años. Con este objetivo nace la agricultura de precisión, persiguiendo la máxima optimización de los recursos del campo.

Mediante la agricultura de precisión se busca que cada explotación agrícola se gestione de manera personalizada, exprimiendo así al máximo los recursos disponibles. Se distinguen cuatro etapas:

- Monitorización.Detección y mapeo de las variables que interesan en cada momento.
- Toma de decisiones. Elaboración de un plan de acción para resolver el problema.
- Actuación. Ejecución del plan de acción.
- Comprobación. Evaluación de la rentabilidad, tanto económica como medioambiental, para aplicarlas al plan del año siguiente.

El uso de UAV se centra en la etapa de monitorización. El método usado anteriormente es el uso de imágenes tomadas por satélite. Este método tiene dos problemas principales: No pueden trabajar en días nublados y las imágenes tomadas no tienen la suficiente resolución. El uso de UAV mejora este método y aplica soluciones a los problemas anteriores.

Algunas de las aplicaciones más usadas de los UAV en la agricultura de precisión son:

- Detección de áreas de infestadas por malas hierbas.
- Detección de zonas que necesitan mayor o menor riego.
- Detección de plagas de hongos e insectos.

Para profundizar en este tema pueden consultarse los artículos citados en la bibliografía [16][18] de este informe.

Aplicación de UAVs en el mantenimiento de tendido eléctrico

La electricidad es el motor de la sociedad moderna y se necesita en cualquier lugar del mundo civilizado. El uso de tendidos de alta tensión aéreos son la forma más común de transportar la electricidad desde las estaciones eléctricas a las urbes.

Hasta hace unos años el mantenimiento de estos tendidos era bastante costoso y arriesgado. Para poder inspeccionarlos, un helicóptero con una cámara se acerca a los tendidos eléctricos. Para evitar el costoso precio de las reparaciones en caso de accidente y, sobre todo, para evitar arriesgar vidas humanas se han empezado a introducir el uso de UAV para este tipo de trabajos.

El uso de UAV en el mantenimiento en tendidos eléctricos de alta tensión se centra sobre todo en los siguientes campos:

- Inspección intensiva de líneas. Esta acción la realiza un operario que recorre el tendido eléctrico. El fin del uso de un UAV para hacer este trabajo es evitar el riesgo de caída del operario.
- Inspección aérea de líneas. Se usan helicópteros para esta acción. El uso de los UAV minimiza el riesgo de dañar el tendido eléctrico y también disminuye los costos de mantenimiento.

- Topografía. El fin es sustituir los helicópteros por UAV para la toma de imágenes. Otro objetivo es servir como apoyo a los topógrafos a nivel de suelo.

- Apoyo en caso de emergencia. En la actualidad se usan helicópteros en caso de avería grave. El uso de UAV puede optimizar el tiempo de respuesta ante una avería y ayudar a resolverla lo antes posible.

- Transporte de cargas. Con el fin de hacer lo más cómodo posible el trabajo de los operarios en los tendidos aéreos, los UAV pueden transportar herramientas y piezas de recambio al operario.

- Tendido de cables. Para instalar los cables de alta tensión entre dos postes es necesario instalar un cable guía para poder desplazar por él después el cable principal, de mayor peso. El uso de UAV para llevar de un poste a otro el cable guía se está implementando para salvar accidentes geográficos complicados (barrancos, ríos, etc).

En España, las principales compañías eléctricas están empezando a implementarlos. Para desarrollar más la información, se puede consultar el artículo[11] que aparece en la bibliografía.

2.2. STRUCTURE FROM MOTION

El término Structure from Motion (SfM) define un sistema de obtención de puntos en 3D.El objetivo es la obtención de la posición en 3D de los puntos característicos de un par de imágenes de un mismo elemento. Una definición aproximada de Structure from Motion es:

> El algoritmo Structure from Motion (SfM) se define como la estimación de la estructura 3D de un objeto rígido y el movimiento relativo de la cámara entre imágenes 2D, cuando los parámetros externos son desconocidos pero se trasladan[5].

Dicho de otro modo, este método está basado en la obtención de la translación y la rotación de un punto entre dos imágenes. Dichas imágenes se toman con una cámara monocular que se va trasladando. Con estos datos se consigue la localización del punto en el espacio que puede ser representado con fórmulas de reconstrucción 3D basadas en la triangulación, que será desarrollada en profundidad más adelante.

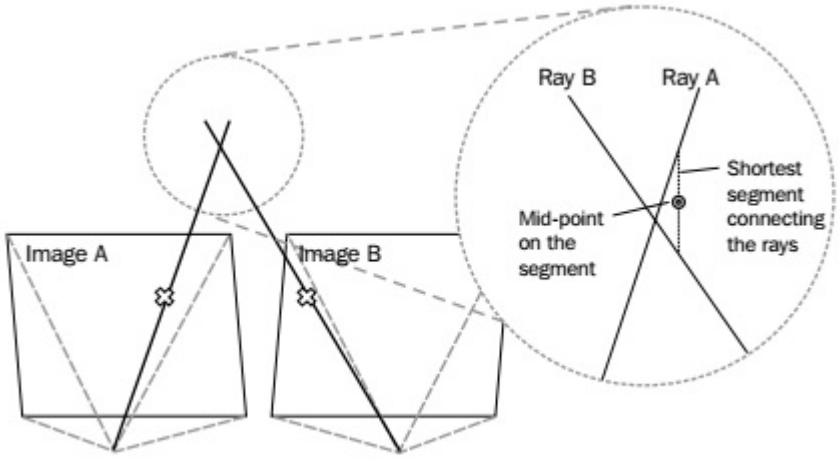


Figura 2.9: *Intersección de puntos característicos de dos imágenes para obtener el punto real en 3D*

Como se aprecia en la figura 2.9, se disponen de dos puntos, uno situado en la Imagen A y otro en la imagen B. Estos dos puntos pertenecen al mismo punto en el espacio, pero el punto de B respecto de A ha sufrido una traslación y una rotación. Si trazamos una línea recta que pase por los puntos focales de la cámara y por dichos puntos en las imágenes A y B, estas dos líneas interseccionan en un punto del espacio. Este punto es la representación en 3D del punto real.

Debido a que la rotación y la traslación no es estándar y depende de las condiciones de cada par de imágenes, el punto obtenido a través de estas dos líneas y el punto real tienen un pequeño error que distorsiona la calidad de la reconstrucción. Este error tiene que ser absorbido lo máximo posible por el algoritmo de obtención de estos puntos eliminando aquellos que han tenido una rotación o traslación mayor de la querida.

Para poder aplicar el algoritmo hay que definir la siguiente estructura:

1. Extracción de los puntos característicos de las dos imágenes.
2. Comparación entre los puntos obtenidos y extracción de los valores de traslación y rotación.
3. Estimación de la estructura 3D de los puntos.
4. Reconstrucción de la estructura 3D mediante triangulación.

Como puntos negativos de este sistema cabe destacar la baja definición del resultado final de la reconstrucción 3D debido al carácter dinámico, puesto que las distancias entre tomas no son fijas y genera bastante distorsión en comparación con una reconstrucción 3D con cámaras estéreo.

En los siguientes capítulos se expondrán las partes del algoritmo, así como sus definiciones y explicaciones teóricas, y se desarrollará un algoritmo de detección de la estructura 3D y su representación.

2.3. OBTENCIÓN Y DESCRIPCIÓN DE LOS PUNTOS DE INTERÉS

Uno de los aspectos más importantes del análisis de imágenes por computación es la extracción de información de los puntos característicos del ambiente. Su objetivo es encontrar aquellos puntos de la imagen cuya información defina el contenido de ésta. Existen muchos tipos de información que se pueden extraer y sus aplicaciones son variadas.

En este apartado se va a definir el concepto de punto de interés (denominado en inglés keypoint), explicando sus características y clasificaciones. También se explica la forma de almacenar la información de estas características en los descriptores.

Se va a exponer la implicación de los puntos característicos en el algoritmo Structure from Motion y sus formas de uso, así como algunos métodos de obtención de los puntos y de sus descriptores.

Por último, se va a realizar una comparativa de los métodos de obtención expuestos y se va a decidir el que mejor se adapta a las exigencias del proyecto.

DEFINICIÓN DE PUNTO DE INTERÉS DE UNA IMAGEN

La definición de punto de interés tiene su complejidad, dependiendo de la aplicación a la que vaya dirigida la definición puede ser más o menos completa. Como punto de partida, un punto de interés de una imagen contiene características o cualidades (en inglés features) destacables de la imagen. En función del fin que se le quiera dar, estas características pueden variar de un proceso a otro.

La detección de características es una operación de procesado de imagen low-level, es decir, estas operaciones se realizan en primer lugar. El algoritmo aplica una serie de filtros para suavizar la imagen, eliminando ruido e información no válida. Después busca píxel a píxel de la imagen las posibles características y almacena esta información en un vector denominado descriptor.

El objetivo de estos algoritmos es la búsqueda de puntos singulares que definan la imagen. Estos puntos contienen información diferente de los píxeles vecinos, definiendo un cambio de tendencia en la imagen y, por lo tanto, una información valiosa para el tratamiento de imágenes. Los puntos de interés más importantes suelen estar dentro de uno de los siguientes grupos:

- Bordes. Un píxel o conjunto de píxeles forman un borde cuando sirven de frontera entre dos zonas de píxeles con información diferente. Estos puntos tienen un alto gradiente en la imagen, lo que identifica el brusco cambio de tendencia entre dos píxeles.

- Esquinas. Cuando un grupo de píxeles agrupados como bordes dan un brusco cambio de dirección. Este término agrupa tanto las esquinas tradicionales como las zonas donde se aprecia una curvatura significativa, los cuales también son puntos de interés.

- Blobs. Traducido al castellano como gota, este término hace referencia aquellas zonas que, debido a la imposibilidad de ser suavizados por el algoritmo, no son detectados como esquinas o puntos de interés. Sobre todo actúa ante brillos en las imágenes y zonas con demasiada claridad.

- Crestas o arrugas. En ocasiones, la distinción de bordes en zonas con un límite muy arrugado o con picos es difícil de conseguir. Estos puntos se denominan crestas y se consideran bordes.

Una vez definidos los puntos de interés de una imagen se almacenan en vectores de información llamados descriptores. En cada una de las posiciones del descriptor se almacena la información que define a ese punto de interés y su posición en la imagen.

Existen bastantes algoritmos de extracción de características, cada uno con sus virtudes y sus inconvenientes. A continuación se expone una muestra de métodos de extracción de puntos de interés como son el método SIFT, el método SURF, el método FAST y el método FREAK.

2.3.1. MÉTODO SIFT

El término SIFT[17] (Scale-Invariant Feature Transform) hace referencia a un tipo de algoritmo de búsqueda de keypoints. Es desarrollado por David Lowe en 1999 en la universidad British Columbia en Vancouver, Canadá. Está bajo patente en los Estados Unidos desde 2004 para su uso comercial. Se trata de un algoritmo robusto ante:

- Ruido en la imagen.
- Cambio en la escala.
- Cambio en el giro.
- Cambio en la iluminación.

Se trata de un detector con un buen rendimiento y precisión en la búsqueda de puntos de interés. Tiene un tiempo de respuesta aceptable, aunque no es el más rápido. Este método dispone tanto de un algoritmo de búsqueda de puntos de interés como la posibilidad de crear sus propios descriptores.Su gran punto fuerte es que puede generar un gran número de descriptores estables y robustos ante cambios.

El algoritmo de detección se divide en cuatro etapas:

1. Identificación de máximos y mínimos.
2. Filtro de los puntos de interés.
3. Determinación de la orientación.
4. Construcción de los descriptores.

A continuación se va a describir cada una de las etapas.

IDENTIFICACIÓN DE MÁXIMOS Y MÍNIMOS.

Este paso consiste en la aplicación de varios filtros sucesivos para así sacar los valores extremos de los puntos de interés.

El primer paso es determinar la posición y escala de forma repetida a diferentes vistas del objeto. El conjunto de vistas es filtrado entonces por una gaussiana. Los puntos sacados son los máximos y mínimos resultantes de las restas de estos filtros a escalas diferentes.

Se define $L(x, y, \sigma)$ como el espacio escala de la imagen, que resulta de convolucionar la imagen $I(x, y)$ con la Gaussiana de escala variable $G(x, y, \sigma)$ de la siguiente forma:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

Donde la Gaussiana viene definida mediante la expresión matemática:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{\sigma^2}}$$

Para obtener un algoritmo más eficiente que obtenga más puntos de interes estables se realiza convolución basada en la Diferencia de Gaussianas (DoG) en el espacio-escala:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)$$

Donde $D(x, y, \sigma)$ representa la convolución de la imagen con la diferencia de las dos gaussianas en σ y $k\sigma$, con k siendo el factor de separación entre escalas definido por $k = 2^{\frac{1}{s}}$. De esta fórmula podemos apreciar que al final $D(x, y, \sigma)$, que es la DoG, se consigue mediante la resta de dos imágenes en diferentes escalas, ahorrando tiempo de procesamiento.

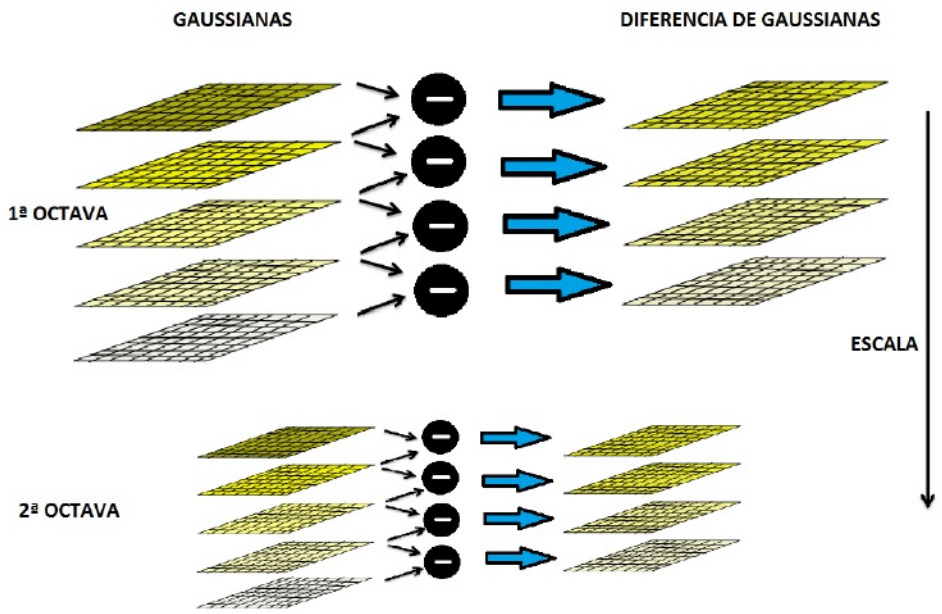


Figura 2.10: *Diferencia de Gaussianas en el escala-espacio*

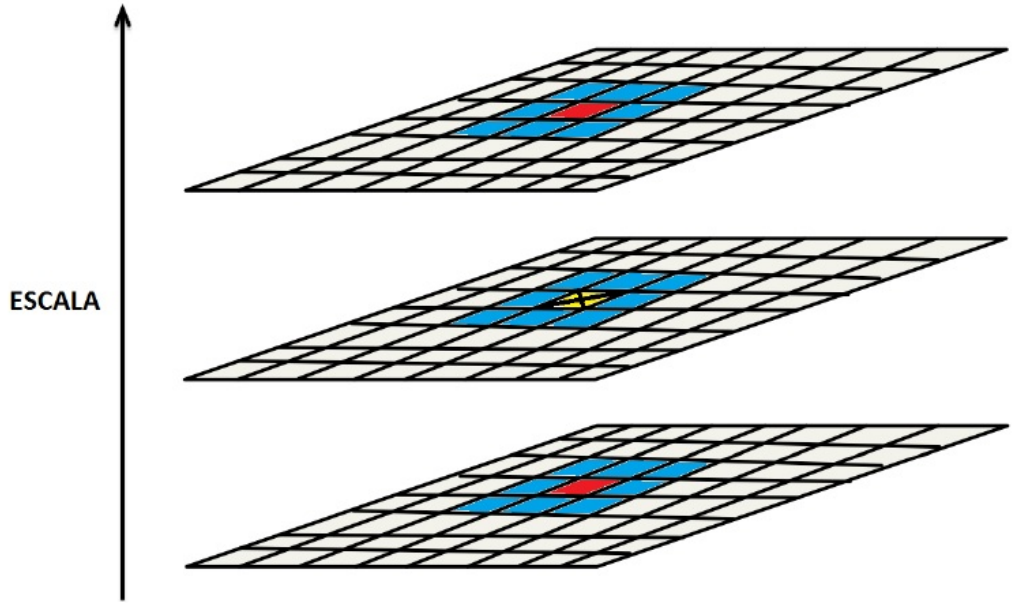


Figura 2.11: *Comparación del punto candidato con sus vecinos entre escalas*

Según expone David Lowe en su artículo, unos valores óptimos para la aplicación de este paso serían: número de octavas = 4, número de niveles de escala = 5 y $\sigma = 1{,}6$.

Los puntos candidatos son máximos y mínimos locales de la DoG a las escalas analizadas. El siguiente paso es comparar cada píxel con sus vecinos adyacentes y también con los nueve vecinos de las escalas superior e inferior. Si el píxel analizado es un máximo o un mínimo, se selecciona como posible punto de interés.

FILTRO DE LOS PUNTOS DE INTERÉS

Se descartan aquellos puntos que han pasado el anterior filtro pero no son adecuados por su localización, contraste, etc. Se establecen varios condicionantes:

- Se interpola cada punto para saber su posición con precisión. Para ello se aplica la expresión de Taylor de la ecuación de la Diferencia de Gaussianas que viene dada por:

$$D(p) = D + \frac{\delta D^T}{\delta p} p + \frac{1}{2} p^T \frac{\delta^2 D}{\delta p^2} p$$

  Donde p es el punto de interés a estudiar. Para hacer las comprobaciones se debe derivar respecto de p y particularizar para $p = 0$. En caso que cualquiera de los valores de la derivada sea mayor que 0.5, el punto de interés será descartado por el algoritmo.

- Para eliminar aquellos puntos con bajo contraste hay que obtener la ecuación de Taylor de segundo orden de $D(x, y, \sigma)$, obteniendo:

$$D(\hat{p}) = D + \frac{1}{2} \frac{\delta D^T}{\delta p} \hat{p}$$

  Si el valor de $D(\hat{p})$ es mayor que 0.03 el punto de interés es descartado.

- Se eliminan los puntos que sean bordes puesto que la función Diferencia de Gaussianas tiene una alta respuesta a los bordes.

Un pico débilmente definido en la función Diferencia de Gaussianas indica una larga curvatura en la dirección del borde pero pequeña en la perpendicular. Esta curvatura se evalúa a partir de matriz hessiana 2x2 evaluada en el punto.

$$\begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}$$

A partir de H se puede obtener la suma de sus autovalores y el producto del determinante:

$$Tr(H) = D_{x,x} + D_{y,y} = \alpha + \beta$$
$$Det(H) = D_{x,x}D_{y,y} - (D_{x,y})^2 = \alpha\beta$$

Donde α es el mayor de los autovalores y β es el menor. Se puede definir r como la relación entre los autovalores $\alpha = r\beta$, relacionando las dos ecuaciones anteriores se obtiene:

$$\frac{(Tr(H))^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$$

Es decir, solo depende de la relación que existe entre los autovalores. Para realizar la relación entre las curvaturas hay que analizar la siguiente ecuación:

$$\frac{(Tr(H))^2}{Det(H)} < \frac{(r+1)^2}{r}$$

Un valor razonable para la relación entre los autovalores sería $r = 10$, por lo que si la anterior ecuación no se cumple se eliminan los puntos de interés.

DETERMINACIÓN DE LA ORIENTACIÓN

En este punto se quiere dotar al punto de interés de una robustez ante cambios en la rotación. Se obtiene el histograma del gradiente de la rotación relativo a los puntos vecinos utilizando la imagen gaussiana de escala más próxima al punto de interés. Para cada imagen muestreada se definen una magnitud $m(x, y)$ y una rotación $\theta(x, y)$:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \arctan \frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}$$

Según el valor el gradiente se pondera a cada vecino y se asocia una gaussiana de $1{,}5\sigma$ con respecto al punto de interés. Las orientaciones dominantes coinciden con los picos de histograma. De esta forma se crea un nuevo punto de interés con dirección relativa al pico máximo del histograma y con cualquier dirección que sea al menos el 80 % de dicho valor.

CONSTRUCIÓN DE LOS DESCRIPTORES.

Una vez en este punto, se tiene una lista de puntos de interés cada uno con su orientación, escala y posición. El último paso es la creación de los descriptores para almacenar la información de estos puntos.

Para la creación de los descriptores el algoritmo hace uso de los histogramas de orientación. Todos los descriptores deben permanecer invariantes ante cambios en la imagen y han de poder ser distinguidos en un par de imágenes.

Para calcular el descriptor se usa el histograma de orientación sobre una región de dimensiones $4x4$ en el entorno del punto. Estos histogramas se construyen en base a la orientación del punto principal y se pondera por la magnitud del gradiente asociado y por una gaussiana $1{,}5\sigma$del punto estudiado.

Las dimensiones del descriptor constituyen un vector de 128 elementos. Para darle robustez frente a cambios de luminosidad se le aplica una última normalización.

Así el descriptor contiene: Posición del punto de interés en coordenadas (x,y), la escala, la orientación y la información del entorno (conjunto de gradientes vecinos).

Para profundizar más acerca de este método de obtención se puede referir a los artículos [17] [12] [3] de la bibliografía.

2.3.2. MÉTODO SURF

El algoritmo SURF (Speeded Up Robust Features) hace referencia a un algoritmo de búsqueda y almacenamiento de puntos de interés. Desarrollado por Herbert Bay [10] y publicado por primera vez en 2006 en la Conferencia Europea de Visión por Computador. Este método de obtención de puntos de interés consta de varias etapas:

1. Identificación de puntos de interés.
2. Determinación de la orientación.
3. Construcción del descriptor.

IDENTIFICIÓN DE PUNTOS DE INTERÉS

A diferencia de otros métodos de extracción de puntos de interés, el método SURF utiliza una aproximación de la matriz Hessiana para determinar la escala y la posición, obteniendo una mayor velocidad de cálculo y también una alta precisión.

Para un punto $p(x, y)$, en la imagen I, la matriz Hessiana a escala σ se describe como:

$$H(p,\sigma) = \begin{bmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{yx}(p,\sigma) & L_{yy}(p,\sigma) \end{bmatrix}$$

Donde:

$$L_{xx}(p,\sigma) = \frac{\partial^2}{\partial x^2} g(\sigma)$$

De manera similar se pueden definir $L_{xy}(p,\sigma)$, $L_{yx}(p,\sigma)$ y $L_{yy}(p,\sigma)$.

Debido a algunas limitaciones de los filtros gaussianos (aliasing, necesidad de discretización, etc) SURF hace uso de filtros de caja, que consiste en una estimación de las derivadas parciales de segundo orden de las gaussianas involucradas.

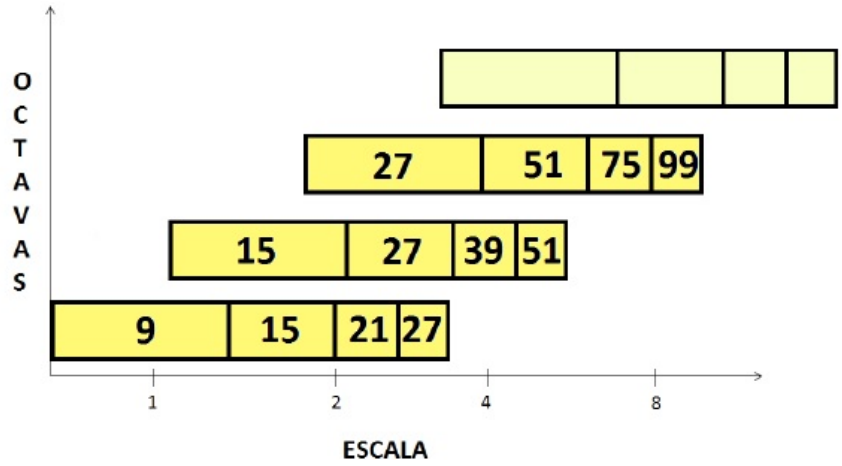


Figura 2.12: *Estructura y tamaño de los filtros utilizados en SURF a diferentes escalas*

Se definen D_{xx}, D_{xy} y D_{yy} como las aproximaciones de las derivadas parciales. El determinante de la matriz Hessiana se puede definir como:

$$det(H_{aprox}) = D_{yy}D_{xx} - (0{,}9D_{xy})^2$$

Gracias al filtro de caja no es necesario aplicar ningún filtro adicional a la salida de éste. Inicialmente se aplica un filtro de caja de dimensiones $9x9$ que obtiene una salida con escala inicial $a = 1{,}2$ correspondiente a una gaussiana de $\sigma = 1{,}2$.

Al igual que en el método SIFT, en el algoritmo SURF el espacio-escala está dividido en octavas. Sin embargo, en el método SURF las escalas tienen un número fijo de imágenes como resultado de la convolución de la misma imagen original con una serie de filtros cada vez más grande.

Para calcular la posición de todos los puntos de interés en todas las escalas se eliminan todos aquellos puntos que no cumplan la condición de un vecindario máximo de $3x3x3$. De esta manera, el máximo determinante de la matriz Hessiana es interpolado en la escala y posición de la imagen.

DETERMINACIÓN DE LA ORIENTACIÓN

Una vez obtenidos los candidatos a punto de interés, el siguiente paso es determinar para cada uno una orientación, haciéndolos invariantes ante cambios en la rotación.

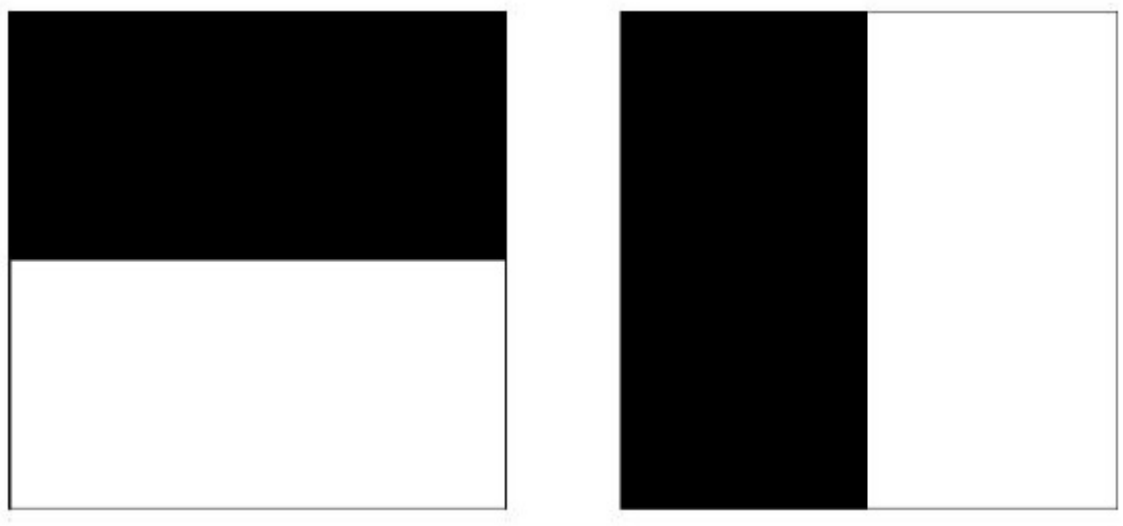


Figura 2.13: *Funciones de Haar para el detector SURF*

Para ello se calcula la respuesta de Harr con dirección en x e y, como aparece en la figura 2.13, en un entorno circular de radio $6s$ en el punto de interés (s es la escala del punto centrado). Las respuestas onduladas varían en función de s, cuanta más escala mayor será la dimensión de la respuesta. Seguidamente se usa de nuevo las imágenes integrales consiguiendo un filtrado más rápido.

Las respuestas obtenidas se ponderan mediante una gaussiana con $\sigma = 2{,}5s$ centrada en el punto de interés. La orientación dominante se obtiene mediante la suma del conjunto de respuestas en una ventana de orientación variable que cubre un ángulo espacial. Este parámetro se obtiene de forma experimental y permite cubrir aproximadamente $\frac{3}{\pi}$ radianes. Se construye un nuevo vector con la suma de las dos respuestas, la vertical y la horizontal, con la orientación determinada por el vector de mayor longitud.

CONSTRUCCIÓN DEL DESCRIPTOR

Una vez se han conseguido los puntos de interés se procede a crear un descriptor para su almacenamiento. Para ello una región cuadrada con la orientación calculada en el paso anterior y centrada en el punto de interés de tamaño $20s$. Se reducen las regiones a subregiones de 4x4 y para cada una de ellas se determinan las características en puntos

diferenciados por regiones de tamaño 5x5. Se definen las respuestas de Haar para cada una de las direcciones, d_x y d_y, referenciadas a la orientación del punto de interés. Estas respuestas se van a ponderar con una gaussiana de $\sigma = 3{,}3s$. Con esta acción se consigue robustez contra deformaciones geométricas y errores de posicionamiento.

Las respuestas d_x y d_y de cada subregión se suman dando la información del descriptor. También se suman $|d_x|$ y $|d_y|$ obteniendo información de la polaridad.

Cada subregión del vector descriptor SURF tiene cuatro conjuntos de información por cada elemento y una extensión de 64 elementos.

Para profundizar más en el tema se pueden consultar los artículos [10][12] de la bibliografía.

2.3.3. MÉTODO FAST

El algoritmo FAST (Features from Accelerated Segment Test) es un método de búsqueda de puntos de interés desarrollado por Edward Rosten y Tom Drummond [20] en 2006. Se trata de un algoritmo de detección de esquinas con un bajo coste computacional, lo que lo hace idóneo para el uso en computación visual aplicado a la robótica.

A diferencia de los métodos SIFT y SURF, el algoritmo FAST no posee un sistema de creación de descriptores, por lo que debe utilizar los descriptores de cualquiera de los dos otros métodos.

DETECTOR DE PRUEBAS SEGMENTO

El detector usa un círculo de píxeles de Bresenham de radio 3 para clasificar si el punto central es una esquina, como se puede ver en la figura. Cada uno de los píxeles de este círculo tiene asignado un número del 1 al 16.

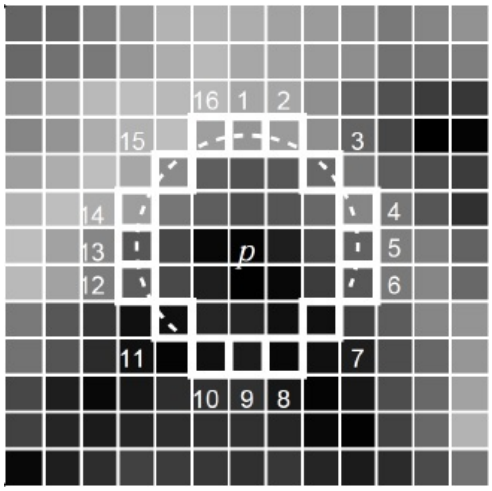


Figura 2.14: *Círculo de Bresenham de radio 3 píxeles con el punto de estudio p en el centro*

Se denomina la intensidad del píxel candidato I_p. Se analiza el brillo del resto de los puntos del círculo y se compara con I_p. El punto candidato será una esquina si cumple alguna de las siguientes condiciones:

- Un conjunto de N píxeles contiguos S, para todo x perteneciente a S, la intensidad del punto x: $I_x > I_p + t$.
- Un conjunto de N píxeles contiguos S, para todo x perteneciente a S, la intensidad del punto x: $I_x < I_p - t$

Para abreviar, en adelante se van a denominar condiciones de umbral. Si alguna de estas dos condiciones se cumple, el punto candidato se considera una esquina. Existe una compensación en la elección de N y t. Este método no es muy preciso a costa de ser muy veloz. Sin la mejora de aprendizaje de la máquina, un valor óptimo de N = 12.

ESTUDIO DE ALTA VELOCIDAD

Para hacer el algoritmo más rápido primero se compara la intensidad de los píxeles 1, 5, 9 y 13 del círculo con I_p. Al menos 3 de los 4 puntos deben satisfacer el criterio del umbral para que exista punto de interés. Si se da esta condición, se analiza entonces que 12 de los 16 puntos del círculo cumplan la condición del umbral.

Sin embargo, este método de ensayo tiene varias debilidades:

1. La prueba de alta velocidad no se puede generalizar bien para $N < 12$. En este caso, puede que sólo 2 de las 4 comprobaciones de brillo de píxeles cumplan las condiciones de umbral.

2. La eficiencia del detector depende de la elección y el orden de los píxeles de prueba seleccionados. Sin embargo, es poco probable que los píxeles seleccionados tengan las condiciones óptimas de distribución de las apariencias de la esquina.

3. Muchas características se detectan unas a otras.

MEJORA DEL ALGORITMO CON EL APRENDIZAJE DE LA MÁQUINA

Para hacer frente a los dos primeros puntos débiles descritos en el apartado anterior se introduce un enfoque de trabajo basado en el aprendizaje de la máquina para ayudar a mejorar el algoritmo de detección. Funciona en dos etapas, la detección de esquinas con un valor N se procesa en un conjunto de imágenes de entrenamiento. Se aplica el algoritmo de la forma más sencilla y se comparan con unos valores de umbral apropiados.

Para el punto candidato p, cada localización del punto se denota como $p \rightarrow x$. El estado de cada píxel $S_{p \rightarrow x}$ debería pertenecer a uno de los siguientes estados:

- d, $I_{p \rightarrow x} \leq I_p - t$ (más oscuro).
- s, $I_p - t \leq I_{p \rightarrow x} \leq I_p + t$ (igual).
- b, $I_{p \rightarrow x} \geq I_p + t$ (más oscuro).

En función de los valores obtenidos se puede dividir el vector en tres diferentes particiones, P_d, P y P_b donde:

- $P_d = p \in P : I_{p \rightarrow x} = d$
- $P_s = p \in P : I_{p \rightarrow x} = s$
- $P_b = p \in P : I_{p \rightarrow x} = b$

En segundo lugar, se declara una variable booleana K_p que indica si p es o no un punto de interés. Se aplica el algoritmo ID3 (algoritmo de árbol de decisión) para consultar cada subconjunto utilizando la variable K_p para conocer cuál es la clase verdadera.

Este algoritmo funciona con el principio de minimización de la entropía. Se utiliza la variable K_p para medir la cantidad de información de p para ser una esquina. Para un conjunto de píxeles Q, la entropía total de K_Q es:

- $H(Q) = (c+n)\log_2(c+n) - c\log_2(c) - n\log_2(n)$
  - Donde c es el número de esquinas (k es verdadera).
  - Donde n es el número de no esquinas (k es falsa).

Se aplica de forma recursiva la minimización de la entropía a los tres subgrupos. El proceso termina cuando la entropía de un subgrupo es cero, de modo que todos los píxeles de ese subgrupo son o no esquinas, es decir, puntos de interés.

Para más información acerca del algoritmo se puede consultar los artículos [20][22] de la bibliografía.

2.3.4. MÉTODO FREAK

El método FREAK (Fast Retina Keypoint) es un algoritmo de reconocimiento de puntos de interés desarrollado por Alexandre Alahi[7] en 2012. El algoritmo pretende imitar el funcionamiento de un ojo humano para la captación de las características del entorno.

A diferencia del resto, este método solo genera descriptores y no tiene un buscador de keypoints, teniendo que utilizar otros como pueden ser los keypoints SURF o SIFT.

El algoritmo consta de cuatro etapas diferenciables:

1. Patrón de muestreo de la retina.
2. Descriptor grueso a fino.
3. Búsqueda del movimiento saccade.

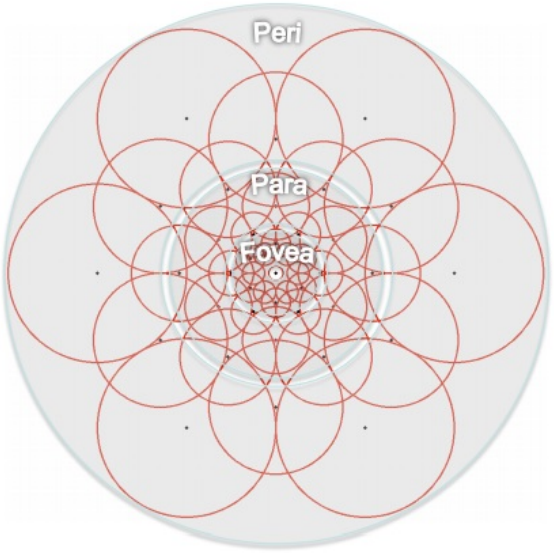


Figura 2.15: *Patrón de análisis de la imagen basado en el funcionamiento de una retina humana. En las zonas centrales se concentra la mayor búsqueda de puntos de interés, donde se aplica la mayor parte de los filtros.*

4. Orientación.

A continuación se va a describir cada una de estas secciones.

PATRÓN DE MUESTREO DE LA RETINA

El método FREAK propone utilizar un método de captación de características de la imagen mediante una imitación de la red de muestreo de la retina del ojo humano.

Antes de comenzar el estudio de los puntos, el algoritmo pasa a cada punto una serie de filtros de diferente tamaño para hacerlo menos sensitivo al ruido. Como se aprecia en la imagen 2.15, cada círculo representa la desviación estándar de los resultados de los filtros aplicados a un punto.

Aumentando el grano del tamaño de la Gaussiana de filtro se producen mejoras de rendimiento. Dentro de este algoritmo, recurren a la redundancia aplicando varias veces el mismo método para conseguir mejores resultados. De esta forma, consideran las intensidades I_i para cada sección A, B y C donde:

$$I_A\, I_B,\ I_B\, I_C \text{ e } I_A\, I_C$$

Si los campos no muestran superposición, el último test $I_A\, I_C$ no añade ninguna información discriminante. Si hay superposición, se puede codificar nueva información.

DESCRIPTOR GRUESO A FINO.

Se construye un descriptor binario F formado por una secuencia de Diferencias de Gaussianas (DoG):

$$F = \sum_{0aN} 2^a T(P_a)$$

donde P_a es el par de campos receptivos, N es el tamaño deseado del descriptor t $T(P_a)$ será 1 si $(I(P_a^{r_1}) - I(P_a^{r_2})) > 0$ y 0 en caso contrario. Muchos de estos pares podrían no ser útiles para describir eficientemente una imagen. Para ello, se toman los siguientes pasos:

1. Se crea una matriz D de casi cincuenta mil puntos de interés extraídos. Cada fila corresponde a un punto de interés junto a su descriptor de todos los posibles pares en el patrón de muestreo.

2. Se calcula la media de cada columna. Como tiene una función discriminante se busca una alta varianza, aceptando un valor de 0.5.

3. Se ordenan las columnas con respecto a la varianza más alta.

4. Se mantiene la mejor columna y de forma iterativa se agregan las columnas restantes de valores inferiores.

Se prefiere la ordenación grueso-fino de la DoF de forma automática. Se captura un esquema debido a la orientación del patrón a lo largo del gradiente global. Este funcionamiento de captura de mayor a menor gradiente es similar al usado por el ojo humano.

BÚSQUEDA DEL MOVIMIENTO SACCADE

Los ojos humanos no están fijos a la hora de captar imágenes sino que se encuentran en movimiento. Cuando el ojo se mueve, el cerebro bloquea el procesamiento de imágenes para volver a iniciarlo cuando se para. A esto se le denomina efecto Saccade.

Este método propone imitar este efecto. Para ello, se comienza a buscar con los 16 primeros bytes el descriptor en la zona de mayor gradiente. Si este valor es mayor que un umbral, se continúa en la zona de menor gradiente. Como consecuencia de este método, se decartan el 90 % de los candidatos en los 16 bytes primeros.

ORIENTACIÓN

Para establecer la rotación del punto de interés, se suman los gradientes locales obtenidos. Sea G el conjunto de todos los pares utilizados para calcular los gradientes locales:

$$O = \frac{1}{M} \sum_{P_o \in G} (I(P_a^{r_1}) - I(P_a^{r_2})) \frac{I(P_a^{r_1}) - I(P_a^{r_2})}{||I(P_a^{r_1}) - I(P_a^{r_2})||}$$

donde M es el número de pares en G y $P_a^{r_i}$ es el vector 2D con las coordenadas espaciales del centro del campo receptivo.

Capítulo 3

EXTRACCIÓN DEL ALGORITMO

3.1. ASPECTOS GENERALES

El objetivo de este capítulo es exponer la estructura general del algoritmo presentado y analizar las estrategias de diseño. En primer lugar se exponen los objetivos y condiciones que debe cumplir el algoritmo. Seguidamente, se muestra la estructura que va a tener. Por último se explicará de forma detallada cada sección del algoritmo. Estas explicaciones disponen del contenido teórico y también de apoyo en forma de artículos y libros para defender las decisiones tomadas.

3.1.1. OBJETIVOS

El objetivo del algoritmo es la extracción de los puntos del entorno para poder representarlos en un entorno 3D. Para ello se va a utilizar el algoritmo de extracción de imágenes a tiempo real Structure from Motion (SfM).

Se puede observar la estructura del algoritmo en el diagrama de flujo de la figura 3.1. El algoritmo es un método iterativo, es decir, desarrolla su acción dentro de un bucle, que puede ser infinito o definido. Dispone de dos lineas de acción que el algoritmo puede tomar:

- Si el algoritmo es ejecutado por primera vez, escoge el camino de la opción SI, asignando a la imagen 2 el valor de la imagen 1.

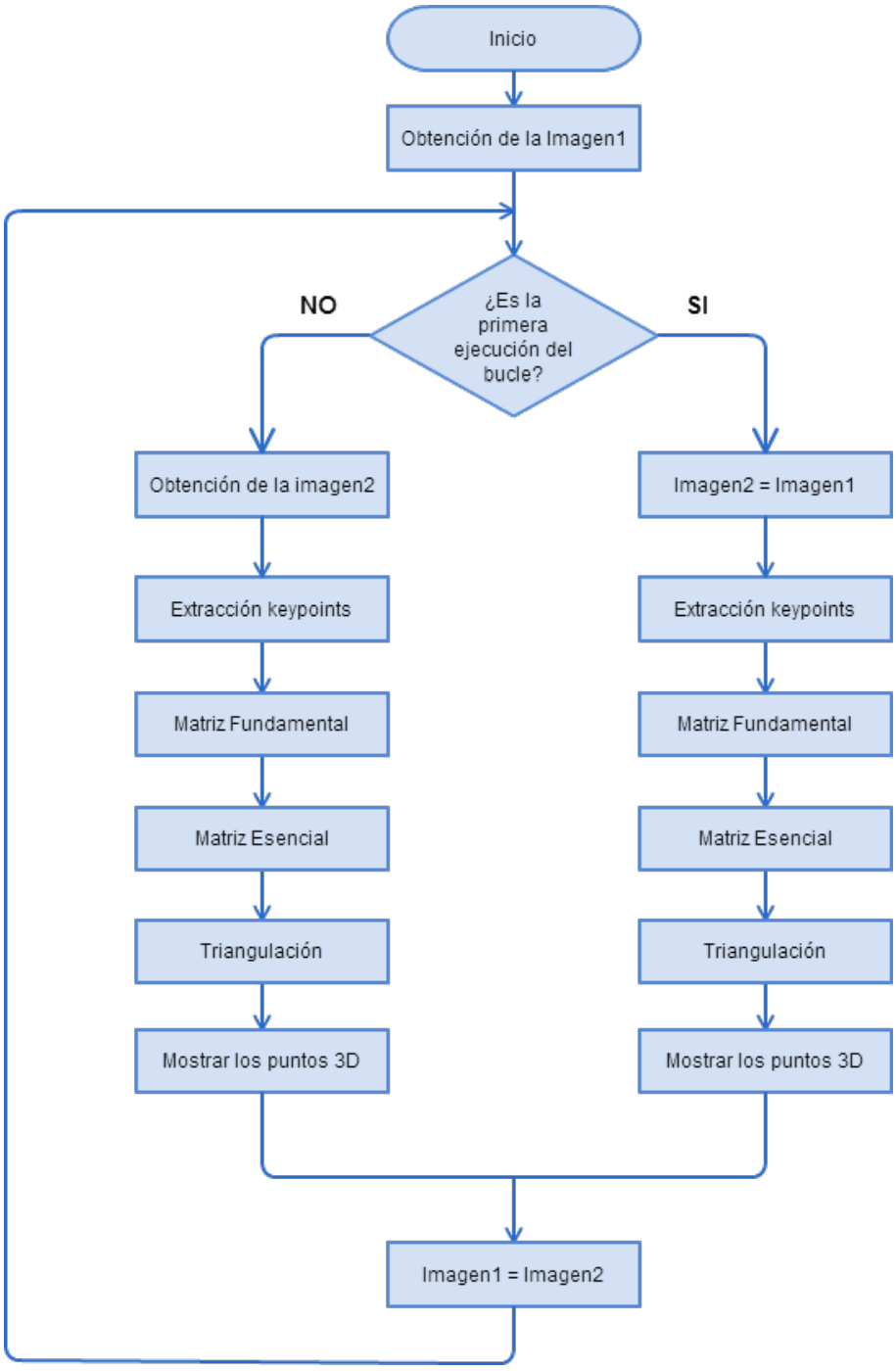


Figura 3.1: *Diagrama de flujo del algoritmo*

- Si el algoritmo es ejecutado de forma continuada, escoge el camino del NO, tomando una nueva imagen 2.

Antes de ejecutarse, tanto la imagen 1 como la imagen 2 no contienen información. Al comenzar la ejecución, la imagen 1 recibe valor, pero la imagen 2 sigue sin contener información. A esta altura del bucle, si se asigna contenido a la imagen 2 coincidirá con la información de la imagen 1 ya que el estado de la captura es el mismo (el bucle no ha avanzado). De esta manera la primera vez se asigna a la imagen 2 el valor de la imagen 1, sabiendo de antemano que la primera comparación tendrá un alto valor de semejanza al final del bucle (teóricamente del 100 % ya que la imagen es la misma).

En el siguiente paso por el bucle, al comienzo se le da un nuevo valor a la imagen 2. En este caso, al comparar la imagen 1 y la imagen 2 no tendremos el 100 % de semejanza ya que la información que almacenan las dos imágenes es diferente. Al final del bucle se asigna el valor de la imagen 2 a la imagen 1. La razón de esta asignación es que en el siguiente paso por el bucle se asignará nueva información a la imagen 2. Al hacer ésto conseguimos que, salvando el caso inicial, durante toda la ejecución del algoritmo se van a comparar dos imágenes obtenidas consecutivamente.

Una vez aclarado este punto, se puede definir el funcionamiento básico de cada parte del algoritmo:

- Obtención de la imagen 1. Se toma una captura por la cámara y se almacena en la imagen 1.

- Obtención de la imagen 2. Se toman valores para la imagen 2. En caso que el bucle se inicie por primera vez se le asigna el valor de la imagen 1, si no es el caso se toma una captura por la cámara y se almacena en la imagen 2.

- Extracción de los keypoints[1]. Se extraen los keypoints de las dos imágenes y se almacenan de sus respectivos descriptores. Se realiza un filtrado para eliminar aquellos

[1] *El término keypoint hace referencia a la forma de nombrar en inglés a los puntos de interés. A partir de este punto y para hacer una mejor identificación se va a usar el término keypoint en sustitución de punto de interés.*

keypoints que no son contenidos en las dos imágenes. A esta acción se la denomina comparación (en inglés matching) de keypoints.

- Obtención de la matriz fundamental. Con los keypoints comparados se construye la matriz fundamental de las dos imágenes que muestra la relación existente entre ellas.
- Obtención de la matriz esencial. Mediante la matriz fundamental y la matriz de calibración de la cámara se obtienen los datos de la matriz esencial, la cual define la rotación y la traslación de los puntos.
- Triangulación. Mediante este método se consigue la posición de los puntos reales con sus coordenadas (x, y, z).
- Representación de los puntos 3D. Se muestran en pantalla los puntos hallados mediante la triangulación.

3.1.2. DATOS DE ENTRADA

El algoritmo hace uso de información del exterior necesaria para poder realizar las operaciones. Las entradas del algoritmo son:

- Imagen 1. Se trata de una imagen en color BGR con un ancho y un alto determinados por la resolución de la cámara. Esta imagen es tomada mediante una cámara y almacenada en esta variable para la comparación con la imagen 2 y poder sacar sus características comunes.
- Imagen 2. Al igual que la imagen 1, se trata de una captura en escala de colores BGR con el mismo ancho y alto que ésta. Es tomada por la misma cámara y almacenada para la comparación con la imagen 1 y su extracción de características comunes.
- Matriz de calibración de la cámara. Matriz de tamaño 3x3 que contiene la información de la distorsión de la lente de la cámara. Esta matriz es necesaria para la obtención de la matriz esencial. Su explicación y valores pueden consultarse en el apéndice 1 - Calibración de la cámara.

3.1.3. DATOS DE SALIDA

La salida del algoritmo será un conjunto de puntos 3D generados mediante triangulación. Este conjunto estará guardado en una nube de puntos y tendrá almacenados las características de los keypoints comunes entre las imágenes 1 y 2, tales como la posición o el color.

3.2. ALGORITMO TEÓRICO

El algoritmo consta de varias etapas denominadas obtención y discriminación de los keypoints, generación de la matriz fundamental, generación de la matriz esencial, triangulación y representación de los puntos.

En el siguiente cuadro se puede ver el algoritmo de iteración y como se accede a cada una de las etapas:

Algoritmo 1 Algoritmo de obtención de puntos en 3D

Entrada: Imagen1, Imagen2.

Salida: Nube de puntos 3D.

1: **mientras** el programa se ejecute. **hacer**
2: Obtención y discriminación de los keypoints.
3: Generación de la matriz Fundamental.
4: Generación de la matriz Esencial.
5: Obtención de los puntos reales mediante triangulación.
6: Representación espacial de los puntos reales.
7: **fin mientras**

El algoritmo 1 representa el funcionamiento de la estructura general de la iteración, independientemente de si el algoritmo se ejecuta por primera vez o no.

En el siguiente algoritmo se va a desarrollar de forma teórica todos los pasos de este algoritmo para su comprensión y su aplicación en cualquier lenguaje de programación.

3.3. DESARROLLO DEL ALGORITMO

Dentro de este apartado se va a explicar de forma detallada el funcionamiento de cada bloque del algoritmo, exponiendo sus fundamentos teóricos y las explicaciones apoyado en bibliografía específica.

3.3.1. OBTENCIÓN Y DISCRIMINACIÓN DE LOS KEYPOINTS

Como primer paso del algoritmo, en esta sección se van a introducir las variables de entrada imagen 1 e imagen 2, las cuales van a ser estudiadas para la extracción de los keypoints. A continuación, serán almacenados en vectores de keypoints y comparados entre sí para descartar aquellos que no coinciden en las dos imágenes.

El funcionamiento de este bloque se puede ver en el siguiente algoritmo:

Algoritmo 2 Algoritmo de extracción de keypoints y su comparación.

Entrada: Imagen1, Imagen2.

Salida: keypoints de la Imagen1 y la Imagen2 en común.

1: Obtención de la imagen 1.
2: Obtención de la imagen 2.
3: Pasar las imágenes 1 y 2 de escala BGR a escala de grises.
4: Extracción de los keypoints de las dos imágenes.
5: Comparación de los keypoints
6: Eliminación de los keypoints que no sean comunes.

OBTENCIÓN DE LAS IMÁGENES.

La parte fundamental del programa es la obtención de imágenes para su estudio. De ellas depende en gran medida que el algoritmo tenga buenos o malos resultados. En este caso, las imágenes serán captadas de forma consecutiva, y entre ellas hay características en común:

- La dos imágenes tienen la misma resolución, es decir, el tamaño de las dos imágenes es el mismo.

- Los efectos de distorsión de la lente de la cámara será el mismo para las dos imágenes.
- La imagen 2 sufre una rotación y una traslación respecto de la imagen 1 como consecuencia del desplazamiento de la cámara.
- Las dos imágenes son tomadas en una escala de color. La escala más común es BGR.

Hay que tener en cuenta que ningún extractor de keypoints puede obtenerlos con una imagen a color. Para conseguirlos hay que pasar la imagen a nivel de grises.

(a) Imagen en escala BGR (b) Imagen en escala de grises

Figura 3.2: *Tipos de escalas de imagen. En el caso (a) se encuentra la imagen en escala BGR. En el caso (b) se encuentra la misma imagen pero en este caso en escala de grises.*

Se establece una escala de 256 tonalidades de gris, desde el blanco al negro. Cada uno dispone de un número dentro de esta escala, variando desde el blanco al negro. Cada tonalidad de gris se asigna a un color de la imagen BGR. Se toma el valor del color BGR de cada píxel, se busca la equivalencia en la escala de grises y se representa en la misma posición en la imagen en escala de grises.

OBTENCIÓN DE LOS KEYPOINTS.

Con las imágenes en escala de gris se puede proceder a buscar los keypoints de las imágenes. Para ello, es necesario crear dos vectores donde se van a guardar. Estos vectores

tienen formato de keypoint y una dimensión igual al número de ellos que contenga cada imagen. De igual forma se crean los descriptores de cada imagen donde se va a almacenar la información del entorno de cada keypoint en la imagen.

Puede usarse cualquier método de extracción de keypoints que se ha expuesto en este documento, los métodos SIFT, SURF, FAST o FREAK. En todos ellos se usa el mismo funcionamiento:

1. Se crean los vectores de keypoints donde se van a guardar.
2. Se crean los descriptores de cada keypoint.
3. Se extraen los keypoints de la imagen 1 y de la imagen 2 y se almacenan en sus respectivos vectores.
4. Se guarda la información del entorno de cada keypoint en su descriptor.

Con este método se consigue extraer la información de los puntos con características interesantes de cada imagen. Estos puntos almacenan información como la posición en la imagen, su color, su intensidad, etc. Además se guarda también la información del entorno de cada punto, donde se puede ver cómo interactúa con los puntos vecinos.

COMPARACIÓN DE LOS KEYPOINTS.

Una vez extraídas las características que definen a cada imagen hay que ver cuántas son compartidas entre ellas. Si dos keypoints coinciden en un alto porcentaje en su información y posición, es probable que estos dos keypoints pertenezcan al mismo punto real. Aunque existe una comprobación que aclara aún más esta situación. Si los puntos en ambas imágenes comparten mucha información y el comportamiento con los keypoints vecinos es similar en las dos imágenes, se podrá certificar con mayor certeza que dichos puntos pertenecen al mismo punto final.

Esta sección se ocupa precisamente de esto. Para ello se utilizan algoritmos de comparación especializados en esta tarea, como puede ser el algoritmo BruteForce. El funcionamiento de estos algoritmos se basa en las siguientes premisas:

1. Se crean dos vectores, uno para guardar las distancias entre keypoints y otro para guardar los valores obtenidos de dichas comparaciones.

2. Para cada uno de los keypoints de una imagen se hace una comparación con los de la otra imagen. Se busca que haya una coincidencia de estos puntos en las dos imágenes.

3. En caso de que esta coincidencia exista, se evalúan los puntos vecinos. Se toman las dos distancias mínimas a los vecinos y se almacenan en el vector de distancias. De forma paralela, se almacenan los valores de la información de dicha comparación en el vector de información.

Cuando se completa este algoritmo de comparación se dispone de bastante información: la información de los keypoints, tanto de la imagen 1 como de la imagen 2, la información relativa a la comparación de las dos imágenes y las distancias mínimas a los vecinos con información relevante.

El siguiente paso del algoritmo es la agrupación de los elementos con información relevante y la eliminación de los puntos que no aportan. El objetivo de este paso es la optimización del tiempo de procesamiento y la mejora de la calidad de la reconstrucción. Se optimiza tiempo ya que, a la hora de estudiar la rotación y la traslación o de representar los puntos reales, el algoritmo dispone de bastantes menos puntos que estudiar, disminuyendo entonces el tiempo de procesamiento de forma considerable. También se mejora la calidad de la reconstrucción puesto que se eliminan puntos sin información relevante, lo que provocaría una distorsión de los datos finales y un falseamiento de los puntos reales representados.

Para este paso es necesario hacer una criba con la información recibida de la comparación. El proceso es el siguiente:

1. Para poder realizar la eliminación se evalúan las distancias obtenidas en la información. Se busca que el cociente entre estas dos distancias mínimas sea menor que un ratio fijado por el programador. De esta forma se puede afinar aún más la búsqueda de los keypoints coincidentes en las dos imágenes.

2. Si la comparación es satisfactoria el algoritmo pasa por este punto sin hacer nada. Si no es satisfactoria, modifica el valor de la comparación dándole un valor negativo. De esta forma lo que se consigue es eliminar el valor de comparación de ese keypoint y no se tendrá en cuenta en futuras operaciones.

3. Por último, se agrupa la información filtrada y se pasa a la siguiente etapa del algoritmo.

El resultado de este algoritmo son los keypoints comunes en las dos imágenes que cumplen un gran nivel de semejanza. Con estos valores se puede buscar la matriz fundamental para seguir avanzando con la búsqueda de los puntos reales.

Para profundizar más en este apartado puede recurrirse a los artículos [8] [21] que aparece en la bibliografía, donde se expone el planteamiento de estos algoritmos de manera clara con ejemplo de aplicación.

3.3.2. GENERACIÓN DE LA MATRIZ FUNDAMENTAL.

Mediante este paso se busca obtener la matriz fundamental, necesaria para la obtención de la matriz esencial. La matriz fundamental ofrece datos de correlación entre los puntos en dos imágenes. Antes de elaborar el algoritmo de esta parte del código se va a definir los conceptos de línea epipolar, la matriz fundamental y también se explicará de forma resumida el funcionamiento del método RANSAC de eliminación de puntos exteriores.

LUGAR EPIPOLAR DE UN PUNTO.

Se define la geometía epipolar entre dos imágenes como la intersección de los planos de las imágenes con los haces de los planos que tienen la línea base como ejes [14].

Para entender de forma más precisa hay que fijarse en la imagen 3.3. Los puntos C situados a ambos extremos de las imágenes son los puntos focales de las cámaras. Estos puntos definen el centro focal de la lente de la cámara, se puede hacer una aproximación

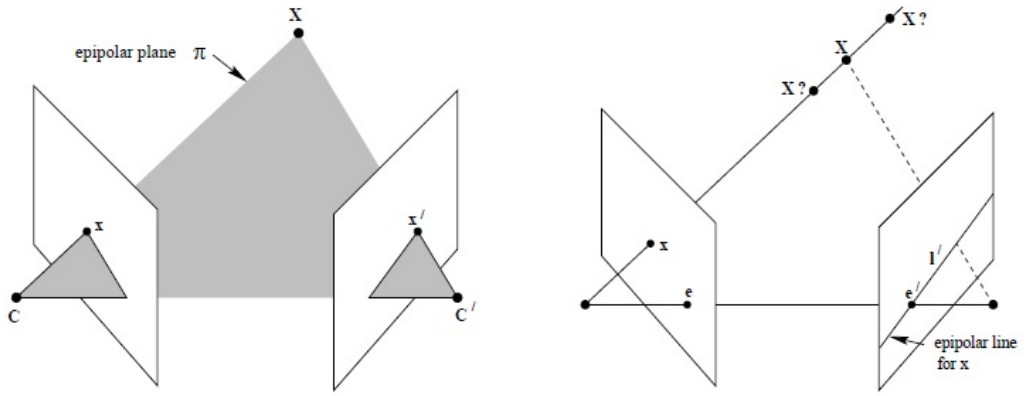


Figura 3.3: *Definición gráfica de lugar epipolar.*

de este punto al centro de la toma de imágenes de cada cámara. los puntos x y x' son los puntos de la imagen real en la imagen 1 y 2 respectivamente y el punto X, el punto real donde los haces de los puntos interseccionan. Si se toma la línea que une los puntos focales de las imágenes (denominada línea base) y las líneas que unen el punto X con los centros de cada cámara (y que pasan por x y x') se forma un plano π llamado plano epipolar.

De igual modo, la línea que resulta de la intersección de cada imagen con el plano epipolar se denomina línea epipolar y a los puntos e y e' epipolos.

De esta manera, se puede definir cada uno de los elementos de la geometría epipolar como:

- El epipolo [14] es el punto de intersección entre la línea base con la imagen del plano. De forma equivalente, el epipolo es la imagen en una vista del foco de una cámara sobre la otra. Este punto da una idea de la dirección del desplazamiento sobre la línea base.

- Un plano epipolar [14] es el plano que contiene a la línea base y el haz del punto real con cada imagen.

- Una línea epipolar [14] es la intersección del plano epipolar con la imagen. Todos los planos epipolares intersectan en el epipolo. Estos planos intersectan tanto a la derecha y definen la correspondencia entre líneas epipolares.

DEFINICIÓN DE MATRIZ FUNDAMENTAL.

La matriz fundamental es la representación algebraica de la geometría epipolar. Para cada punto en una imagen existe una línea epipolar correspondiente en la otra imagen. La línea epipolar es la proyección en la imagen dos de la línea que atraviesa el punto x y el punto C. Por lo tanto existe un mapeo de x en la línea epipolar de la segunda imagen:

$$x \mapsto l'$$

Este mapeo es una correlación singular, que es una proyección del mapeo de puntos a las líneas, que es lo que representa la matriz fundamental.

La matriz fundamental puede ser derivada como la proyección de dos matrices de la cámara. x y x', los puntos correspondientes a en un par de imágenes, indican que el producto de la matriz fundamental F y el punto x deben converger en el punto x' de la otra imagen. Matemáticamente esto se expresa como:

$$(x')^T F x = 0$$

Esta relación define que los puntos x y x', para cumplirla, deben ser coplanares. La importancia de esta relación es la descripción de la matriz fundamental sin la relación con las matrices de las cámaras.

Se tienen dos imágenes tomadas con cámaras sin coincidencia de los centros, entonces la matriz fundamental F es la única matriz $3x3$ de rango 2 que satisface para todos los puntos correspondientes $x \leftrightarrow x'$ que $(x')^T F x = 0$.

Esta definición aporta un número de propiedades que se listan a continuación.

1. F es una matriz homogénea de rango 2 con 7 grados de libertad.

2. Si x y x' corresponden a puntos de las imágenes, entonces $(x')^T F x = 0$.

3. Se define $l' = Fx$ como la línea epipolar correspondiente a x y $l = F^T x'$ como la línea epipolar correspondiente a x'.

4. Se define que la relación entre la matriz fundamental y los epipolos es $Fe = 0$ y $F^T e' = 0$.

OBTENCIÓN DE LA MATRÍZ FUNDAMENTAL

Existen varios algoritmos para la obtención de la matriz fundamental [14]. De todos ellos se elige el método basado en algoritmo RANSAC. La razón de esta elección es que se trata de un algoritmo que extrae la matriz de forma automática y que no necesita a priori ninguna otra información.

El método RANSAC es un algoritmo de discretización de valores ajustados a una condición de cercanía a una media. El algoritmo inspecciona todos los puntos, calcula su media, establece un área aceptación y discrimina los puntos. Los valores pueden ser inliers, aquellos que se encuentran dentro de la zona de aceptación del método RANSAC y outliers, aquellos que se alejan de la media. En caso de que el algoritmo detecte un outlier automáticamente lo descarta. Su funcionamiento se basa en las siguientes etapas:

1. Escoge de forma aleatoria a un subconjunto (llamado inliers hipotéticos), del conjunto de valores de los datos observados.

2. Se crea un modelo en el conjunto de inliers hipotéticos.

3. El resto de los datos observados se evidencian al modelo ajustado. Esos puntos que se ajustan al modelo estimado, según a alguna función de pérdida de modelos específicos, se considerarán como parte del conjunto de consenso.

4. El modelo estimado es bueno si se han clasificado suficientes puntos como parte del conjunto de consenso.

5. Finalmente se puede afinar si se vuelve a utilizar con todos los datos del conjunto.

Para aplicar el método RANSAC sólo se utilizan 7 correspondencias de puntos en la obtención de F. La principal ventaja es la obtención de una matriz de rango 2 en vez de tener que forzarla a serlo como ocurre en los sistemas lineales. Otra de las razones de usar 7 correspondencias en lugar de 8 como en los sistemas lineales, es que el número de muestras a evaluar con el fin de garantizar una alta probabilidad de exclusión de puntos fuera de la media es exponencial al tamaño de la muestra.

DEFINICIÓN DEL ALGORITMO

El algoritmo 3 muestra las etapas de detección de la matriz fundamental descritos anteriormente.

Algoritmo 3 Algoritmo de extracción de la matriz fundamental.

Entrada: keypoints de la imagen 1 y 2, vector de comparaciones y vector de distancias.

Salida: Matriz fundamental 3x3.

1: **para** 0 hasta N, donde N se determina de forma adaptativa dentro del algoritmo. **hacer**
2: Selección de 7 correspondencias y computación de la matriz F.
3: Cálculo de la distancia entre correspondencias.
4: Cómputo del número de inliers consistentes.
5: **si** Hay tres soluciones. **entonces**
6: Seleccionar la de mayor número de inliers
7: **fin si**
8: **fin para**

Para ampliar la información de la matriz fundamental se puede acceder a las referencias [14] [8] de la bibliografía.

3.3.3. GENERACIÓN DE LA MATRIZ ESENCIAL.

Esta sección busca la matriz esencial para obtener los datos de la rotación y la traslación de los puntos. Toma los valores de la matriz fundamental y los especifica para la obtención de las coordenadas de las imágenes.

DEFINICIÓN DE MATRIZ ESENCIAL

La ecuación que define a la matriz esencial, en términos de coordenadas normalizadas que corresponden a los puntos $x \leftrightarrow x'$ es:

$$(\hat{x}')^T E \hat{x} = 0$$

Por otro lado también se sabe de [14] que $\hat{x} = k^{-1}x$, por lo que al despejar se obtiene

$$(x')^T(k')^{-T}Ek^{-1}x = 0$$

Si se compara con la relación de la matriz fundamental $(x')^T Fx = 0$, se obtiene la relación con la matriz fundamental que define a la matriz esencial.

$$E = (k')^T Fk$$

Donde F denota la matriz fundamental y k es la matriz de calibración de la cámara, explicada en el Anexo A. Se denota que la matriz esencial es una matriz 3x3, aunque puede ser descompuesta en dos submatrices, una que muestra la rotación y otra la traslación.

$$E = [R|t] = \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{bmatrix}$$

Dispone de 5 grados de libertad, debido a esto tiene más constricciones adicionales que la matriz fundamental. Para obtener las matrices de rotación y traslación se necesita sacarlas de la matriz esencial mediante algoritmo denominado Singular Value Descomposition (SVD).

El método SVD (Descomposición en Valores Singulares en castellano) es una factorización de una matriz real o compleja de tamaño mxn. Mediante este método se pueden sacar los valores de la rotación y la traslación aplicando las siguientes fórmulas.

$$R = UWV_t$$

Donde se denota U como la matriz unitaria de tamaño mxm, V_t es la matriz unitaria de tamaño nxn y W es la matriz ortogonal que tiene los siguientes valores:

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

La variable de traslación t se define como $t = U(0, 0, 1)^T = u_3$, es decir, la última columna de la matriz U.

DEFINICIÓN DEL ALGORITMO

Una vez definida la matriz esencial y su funcionamiento, el siguiente paso es definir el algoritmo para su extracción. Por último se extraerán las matrices de rotación y traslación, que serán mandadas como variable a la triangulación.

Algoritmo 4 Algoritmo de extracción de la matriz esencial y las matrices de rotación y traslación.

Entrada: Matriz Fundamental F y matriz de calibración de la cámara k.

Salida: Matriz de rotación 3x3 y matriz de traslación 3x1.

1: Mediante la matriz de calibración de la cámara y la matriz fundamental se obtiene la matriz esencial 3x3
2: Extracción mediante SVD de las matrices U y V_t.
3: Declaración de la matriz de rotación $R = UWV_t$.
4: Declaración de la matriz de traslación $t = u_3$

Se puede buscar información más a fondo sobre la matriz esencial y también sobre las matrices de rotación y traslación en [14] [8] e información sobre la descomposición SVD en [4], todas pertenecientes a la bibliografía.

3.3.4. OBTENCIÓN Y REPRESENTACIÓN DE LOS PUNTOS REALES POR TRIANGULACIÓN.

Se define el método de la triangulación como la obtención de un punto real X a través de la proyección de los haces que pasan por los dos puntos de imagen, x y x', y los centros de las cámaras C y C' respectivamente.

Como se observa en la imagen 3.4 las proyecciones de los puntos no convergen exactamente en el punto real (es posible que incluso no converjan nunca). Este problema es debido a la aparición de ruido en la imagen que distorsiona los puntos que aparecen en ella. Por lo tanto es necesario buscar una solución a este problema.

En su artículo Richard Hartley y Peter Sturm [14] proponen varios soluciones a este problema.

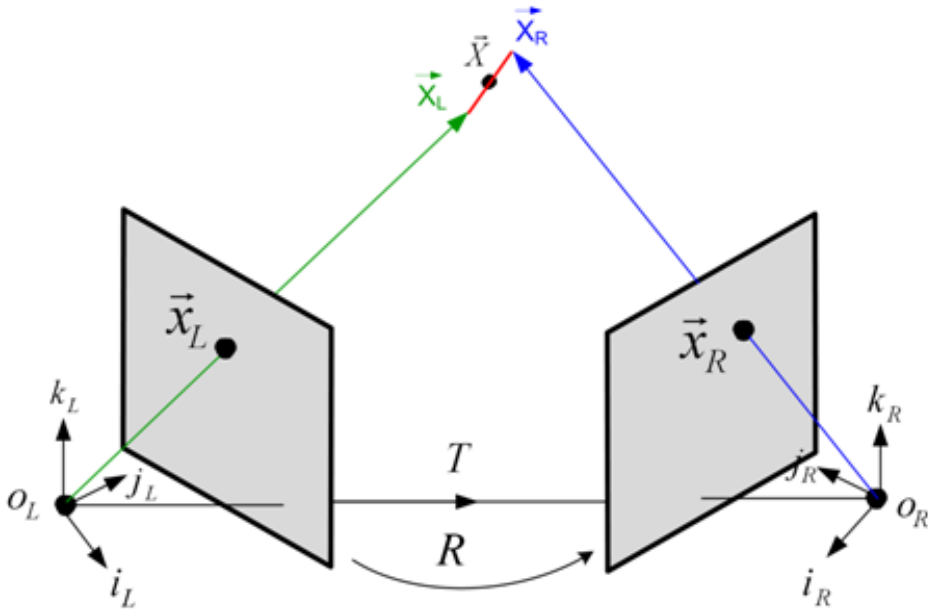


Figura 3.4: *Triangulación de una imagen real. A través de los puntos de las imágenes y los centro focales se puede encontrar la posición real del punto en el espacio*

Los dos haces que corresponden a los puntos de cada imagen interseccionan en el punto real si y sólo si cumplen la siguiente condición:

$$(x')^T F x = 0$$

La desviación de dichos puntos es debida al error que introducen los filtros Gaussianos para filtrar el ruido, por lo que la localización de los puntos exactos puede variar. El fin es identificar de forma clara estas desviaciones y corregirlas, consiguiendo que los haces de los puntos intersecciones en el punto real.

Para poder llevar a cabo la triangulación se elige el método de la triangulación simple LS. Se trata de un método no iterativo que analiza cada par de puntos de la imagen por separado. Aún no siendo el método más exacto, si es el más utilizado debido a su bajo coste computacional comparado con el resto de métodos.

Este método supone $u = Px$, donde u son los puntos la imagen que contienen las coordenadas $w(u, v, 1)$ de dichos puntos, siendo w un escalar, y P es la matriz de traslación y rotación de cada imagen. Para los puntos de la imagen 1 y la imagen 2 se cumple esta

fórmula. Despejando para cada coordenada se tiene las siguientes relaciones:

$$wu = {p_1}^T x \qquad wv = {p_2}^T x \qquad w = {p_3}^T x$$

Eliminando la correspondencia de w mediante la tercera fórmula se obtiene:

$$u{p_3}^T x = {p_1}^T x$$
$$v{p_3}^T x = {p_2}^T x$$

Se obtienen 4 ecuaciones lineales en las coordenadas de los puntos x (2 por cada imagen), que se pueden obtener de la forma $Ax = 0$ para una matriz cuadrada 4x4. Como existen errores debido al ruido de la imagen esta relación no será cero, por lo que se debe calcular el valor para evitar errores.

Por lo tanto, la triangulación se consigue con la resolución de una ecuación no homogénea lineal de la forma $AX = B$.

DESCRIPCIÓN DEL ALGORITMO

Este algoritmo tiene como fin la reconstrucción de los puntos en cada imagen para obtener la posición de los puntos reales del sistema. Para ello el algoritmo 5 explica de forma detallada el funcionamiento de la iteración.

Algoritmo 5 Algoritmo de triangulación.

Entrada: Matriz con las coordenadas reales de los puntos y matriz de rotación y traslación de la cámara.

Salida: Punto real

1: Obtención de la matrix A.
2: Obtención de la matrix B.
3: Resolver mediante el método SVD para obtener X.

El algoritmo necesita las coordenadas (x, y) de cada grupo de vectores. En caso de trabajar con un grupo de cámaras estéreo también necesitaría las matrices de rotación y traslación de cada cámara. Como el informe se desarrolla con un sistema monocular, ambas matrices de rotación y traslación serán la misma, denominada P.

Se necesita resolver la ecuación $AX = B$, por lo que primero de todo se necesita definir las matrices A y B. Ambas matrices se consiguen con las posiciones de los keypoints y la matriz de rotación y traslación P. Sean u, v los vectores que contienen las coordenadas de los keypoints de la imagen 1 y 2 respectivamente, se obtiene entonces que

$$A = \begin{bmatrix} (u.x) * P(2,0) - P(0,0) & (u.x) * P(2,1) - P(0,1) & (u.x) * P(2,2) - P(0,2) \\ (u.y) * P(2,0) - P(1,0) & (u.y) * P(2,1) - P(1,1) & (u.y) * P(2,2) - P(1,2) \\ (v.x) * P(2,0) - P(0,0) & (v.x) * P(2,1) - P(0,1) & (v.x) * P(2,2) - P(0,2) \\ (v.y) * P(2,0) - P(1,0) & (v.y) * P(2,1) - P(1,1) & (v.y) * P(2,2) - P(1,2) \end{bmatrix}$$

$$B = \begin{bmatrix} -((u.x) * P(2,3) - P(0,3)) \\ -((u.y) * P(2,3) - P(1,3)) \\ -((v.x) * P(2,3) - P(0,3)) \\ -((v.y) * P(2,3) - P(1,3)) \end{bmatrix}$$

Las matriz A tiene unas dimensiones 4x3 y la matriz B unas dimensiones de 4x1. Como resultado de esta operación se obtiene un vector X de dimensiones 3x1. Los valores del vector son obtenidos mediante SVD o despejando:

$$AX = B$$
$$A^{-1}(AX) = A^{-1}B$$
$$(A^{-1}A)X = A^{-1}B$$
$$IX = A^{-1}B$$
$$X = A^{-1}B$$

Una vez definido el algoritmo de obtención de los puntos reales se puede definir un algoritmo para representarlos. Como el método de triangulación simple SL no es un método iterativo se genera un algoritmo con iteración que obtenga las variables necesarias para la reconstrucción. Este algoritmo hará uso de la triangulación para acto después representar los puntos.

Algoritmo 6 Algoritmo de representación de los puntos reales.

Entrada: Matriz de rotación y traslación P, vectores de keypoints de las imágenes 1 y 2, matriz de calibración de la cámara.

Salida: Matriz de puntos reales, representación en el espacio.

1: **para todo** los keypoints comunes a las dos imágenes **hacer**
2: Obtención de u y v.
3: Triangulación SL Simple.
4: Almacenamiento del punto real en una nube de puntos.
5: **fin para**
6: Representación de la nube de puntos.

El algoritmo 6 muestra la forma de representar los puntos reales en el espacio. El algoritmo recoge los valores de los keypoints de las imágenes 1 y 2 y los almacena en los vectores u,v de la siguiente manera:

$$u = K(keypoint1.x, keypoint1.y, 1)$$
$$v = K(keypoint2.x, keypoint2.y, 1)$$

Para garantizar que las coordenadas almacenadas en u,v están normalizadas se multiplican por la matriz de calibración de la cámara K.

Una vez se a producido la triangulación los puntos son almacenados en una nube de puntos. Este elemento almacena la información de los puntos reales para que puedan ser representados. Cuando la iteración se ha completado, se puede proceder a representar los puntos reales en el espacio, resultando una imagen en 3D con los puntos obtenidos.

Para profundizar más en este tema se puede acudir a los artículos [8] [15] [14] de la bibliografía de este documento.

Capítulo 4

RESULTADOS PRÁCTICOS

En este capítulo se van a abordar las elecciones y demostraciones que se han ido desarrollando a lo largo del documento. Se va a componer de dos bloques diferenciables. El primer bloque corresponde a las justificaciones de los modelos utilizados para la realización del trabajo. El segundo bloque va a mostrar los resultados obtenidos de la aplicación del algoritmo desarrollado previamente.

4.1. INTRODUCCIÓN

Hasta ahora se ha mostrado un algoritmo de captura de imagen y representación de puntos reales del entorno basado en Structure from Motion. El algoritmo ha sido desarrollado de forma teórica, exponiendo los fundamentos necesarios para poder desarrollar el código en cualquier lenguaje de programación.

Para argumentar el funcionamiento se ha desarrollado un programa basado en el algoritmo con el que conseguir la representación de los puntos reales. Esta sección va a mostrar varias etapas.

- Elección del método de extracción de los keypoints. Se van a mostrar los puntos a favor y en contra de cada uno de los métodos explicados en el informe y se hará una elección según convenga. Se mostrarán ejemplos para respaldar dicha elección.
- Demostración de la obtención de los puntos reales y su representación. Se mostrarán

ejemplos de la representación de los puntos.

Para desarrollar el algoritmo se ha elegido el lenguaje de programación C# desarrollado en el entorno de programación Visual Studio 2013. Se ha elegido éste porque es el lenguaje de desarrollo de los UAV en el departamento de Sistemas. Desarrollando el código en este lenguaje hará más fácil su uso en futuras investigaciones.

Para el desarrollo de la visión por computadora se han utilizado las librerías de tratamiento de imagen Emgu CV, que son una adaptación de las librerías de tratamiento de imagen Open CV para C#. Se trata de un código de programación opensource, facilitando el acceso a ellas.

4.2. ELECCIÓN DEL MÉTODO DE OBTENCIÓN DE KEYPOINTS

Se dispone de varios métodos de extracción de keypoints explicados en este documento. Los métodos SIFT, SURF y FAST son capaces de extraer los keypoints mientras que los métodos SIFT, SURF y FREAK sirven para extraer los descriptores de la imagen.

Para elegir alguno de los métodos de extracción de keypoints anteriores hay que exponer las ventajas e inconvenientes de cada método.

- Método SIFT. Se caracteriza por su buen rendimiento, precisión y tiempo de cálculo. Se trata de un método robusto ante cambios escala, rotación, iluminación y aparición de ruido.

- Método SURF. Muy robusto ante cambios en la imagen. Reducido tiempo de procesamiento, menor que el método SIFT.

- Método FAST. Método bastante rápido en tiempo computacional, es un buen detector de esquinas. Como contrapunto a los anteriores métodos, no dispone de un control de la orientación.

Para poder analizar cada uno de los métodos y compararlos se debe hacer una valoración de cada método para una misma imagen.

El siguiente test de imagen consiste en la generación de los keypoints de una imagen mediante los métodos SIFT, SURF y FAST. Se van a mostrar el tiempo de ejecución, el número de keypoints y su posición en la imagen.

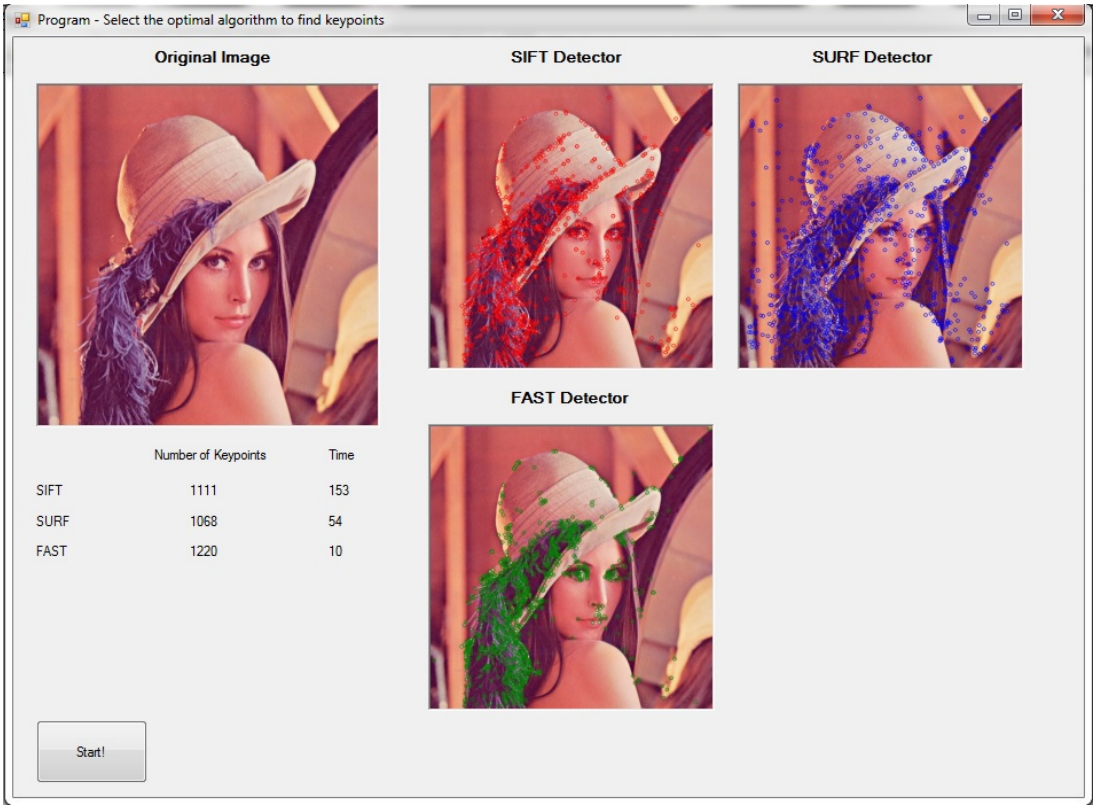


Figura 4.1: *Comparación de los métodos de extracción de keypoints con detalles bien definidos.*

La representación de los keypoints en las imágenes difiere del método de selección de keypoints. En el método SIFT se observa que los keypoints aparecen definiendo la figura de la mujer, destacando mucha información del rostro y el sombrero. El método SURF muestra un número similar de keypoints, sin embargo los puntos están mucho más dispersos y en zonas que no definen la forma como son el fondo. Por último, el método FAST genera un número similar de keypoints que los anteriores, aunque su localización se centra en zonas como los ojos o los detalles del sombrero, pero no definen zonas más

amplias como la cara.

En la siguiente imagen, de mayor detalle en general, se van a aplicar los mismos análisis que en la anterior.

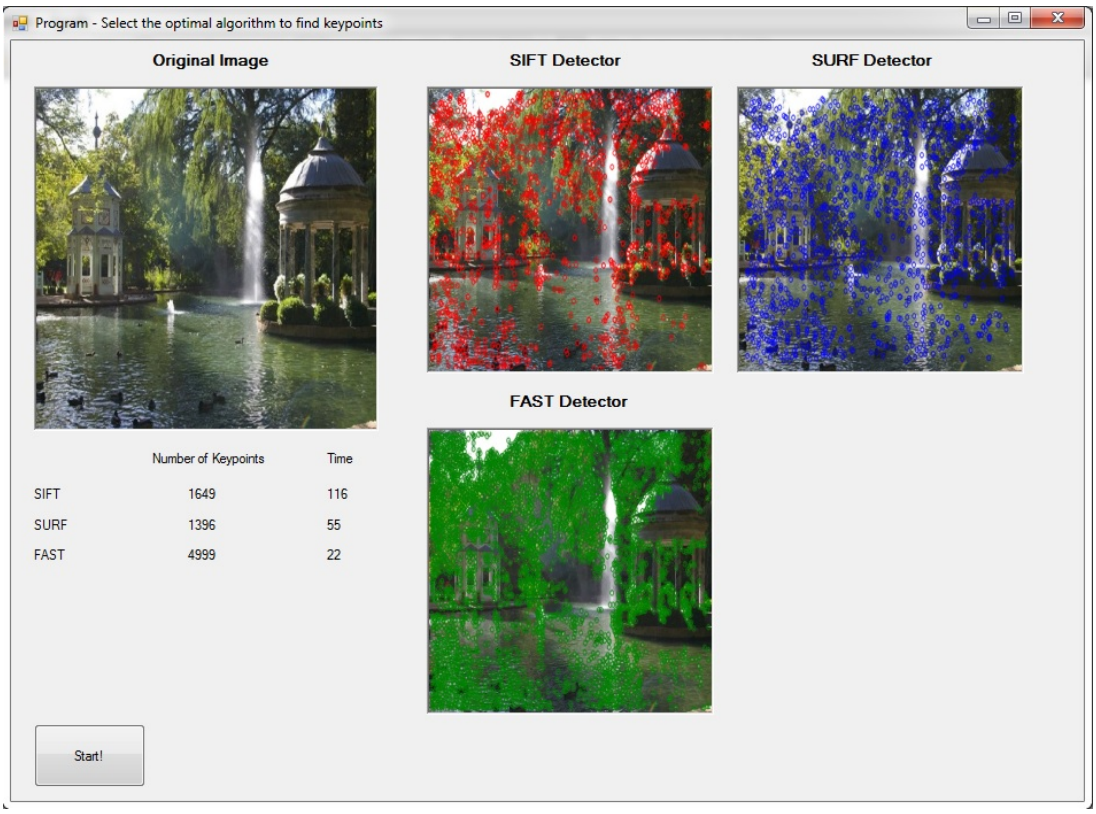


Figura 4.2: *Comparación de los métodos de extracción de keypoints sobre una imagen con muchos detalles.*

La segunda imagen representa un paisaje, con muchos más detalles que en la imagen anterior. En el método SIFT, los keypoints aparecen de forma regular donde se distinguen figuras, definiendo en cada momento dónde hay un objeto. Respecto del método SURF, aunque el número de keypoints es ligeramente menor se puede ver que aparecen en la imagen de forma dispersa, sin definir ningún elemento. Por último, el método FAST aporta muchos más keypoints que el método SIFT, pero presenta mucha saturación de información.

Con estas dos imágenes se puede sacar en claro varios conceptos:

- Frente a imágenes con poco detalle los detectores SIFT y FAST funcionan bien, con

Tabla 4.1: *Datos de análisis de extracción de keypoints de la imagen 1*

Algoritmo	***Número de keypoints***	***Tiempo de procesamiento (ms)***
SIFT	*1111*	153 *ms*
SURF	*1068*	54 *ms*
FAST	*1220*	10 *ms*

una pequeña mejora del método SIFT. El método SURF define las figuras, pero también muchos puntos que no son representativos, lo que hace que el análisis tenga menos calidad que los anteriores.

- Frente a imágenes con muchas figuras y detalle los detectores SIFT y FAST definen bastante bien las figuras. Aunque mientras el método SIFT tiene un buen balance entre cantidad de keypoints y definición, el método FAST genera muchos más keypoints para definir las mismas imágenes. Este método presenta saturación de información, que aumenta el tiempo de procesado durante el algoritmo. Por último, el método SURF presenta los mismos problemas, dispersión de los keypoints que no terminan de definir las imágenes y crea puntos sin información relevante.

En este punto, el método SIFT aventaja al FAST ya que necesita menos keypoints para definir de forma óptima la imagen. Ambos métodos aventajan al SURF debido a la poca definición de los puntos que genera.

Otro punto clave es el tiempo de computación de la extracción de keypoints. En la tabla 4.1 se pueden ver los datos del número de keypoints y del tiempo de ejecución de cada proceso por imagen.

Se aprecia que el tiempo de computación del método SIFT es claramente más alto que el resto de los métodos. En el caso del método SURF, el tiempo es entre 2 y 3 veces más alto. En el método FAST la diferencia es aún mayor, sobre 15 veces mayor.

En la segunda imagen, los tiempos de procesamiento siguen siendo superiores en el método SIFT. Pero mientras que en el método FAST el tiempo aumenta, en el método SIFT disminuye, quedando bastante más igualado que el caso anterior. En el método

Tabla 4.2: *Datos de análisis de extracción de keypoints de la imagen 2*

Algoritmo	***Número de keypoints***	***Tiempo de procesamiento (ms)***
SIFT	*1649*	116 *ms*
SURF	*1396*	55 *ms*
FAST	*4999*	22 *ms*

SURF no varía mucho.

Se ve que respecto al tiempo el método SURF y el método FAST superan con holgura al método SIFT. Aunque es cierto que la diferencia no es crítica, sí que es reseñable indicarla.

Por lo tanto, el método es elegido entre los métodos SIFT y FAST, quedando el método SURF eliminado por sus limitaciones. Entre estos dos métodos, aunque el FAST es más veloz, el SIFT tiene bastante más calidad de obtención de keypoints.

Por lo tanto es entendible sacrificar velocidad de procesamiento por aumentar la calidad imagen. El método de extracción de keypoints será entonces el método SIFT.

En cuanto al método de elección de los descriptores, los posibles candidatos son los descriptores SURF, SIFT y FREAK. Para este análisis hay que recurrir al trabajo de A. Kaff [6] donde hace una comparación entre estos métodos.

En este artículo expone unas comparaciones entre la obtención de los keypoints y descriptores mediante el método SIFT, mediante el método SURF, mediante los keypoints del método SIFT y los descritores FREAK y mediante los keypoints del método SURF y los descriptores FREAK.

En este artículo hace varias comparaciones, respecto del tiempo de procesamiento la combinación SIFT-FREAK reduce al 50 % el tiempo de procesamiento respecto del método SIFT, mientras que la combinación SURF-FREAK apenas reduce el tiempo al método SURF.

Los detectores SIFT y SURF tienen problemas ante los efectos del ruido los cuales son en parte solucionados si se aplican las soluciones SIFT-FREAK y SURF-FREAK.

Mediante la fórmula SIFT-FREAK se puede reducir el efecto del ruido en un 91 % mientras que el efecto del SURF-FREAK se reduce apenas en un 11 % según [6].

Finalmente, respecto a la desviación de método, la fórmula SIFT-FREAK tiene una relación de desviación 0,0, lo que significa que es invariable en contra de la transformación de la imagen (ruido, rotación y escala), mientras que la fórmula SURF-FREAK es muy sensible ante cualquier pequeño cambio en la imagen.

Como conclusión en el artículo se expone que el método SIFT-FREAK es la mejor elección para el uso de sistemas de navegación en tiempo real, en términos de tiempo y exactitud [6].

4.3. REPRESENTACIÓN DE LOS PUNTOS REALES

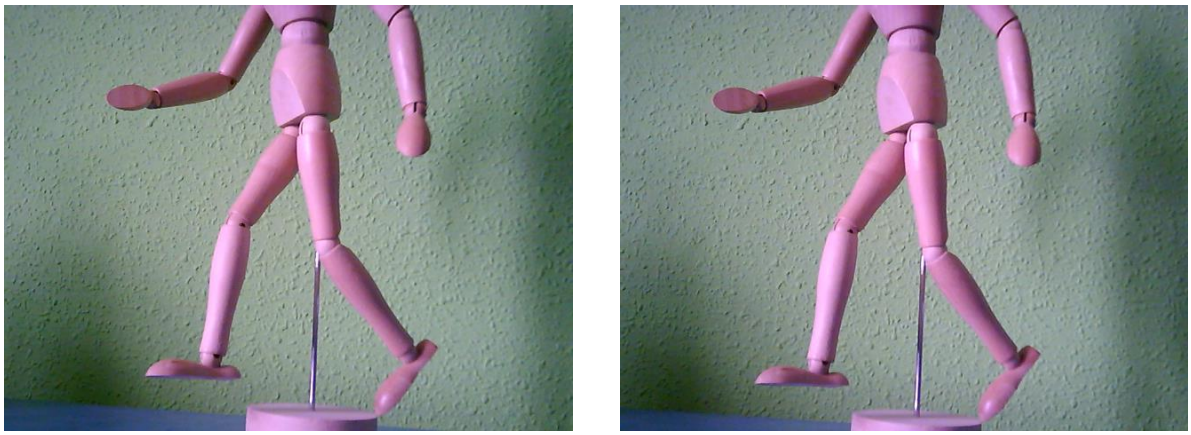
La reconstrucción 3D de los puntos estudiados indica la posición en la que se encuentra el objeto en el espacio. Con el objetivo de evaluar la reconstrucción de los puntos, se plantean varios escenarios en los que se llevará a cabo una reconstrucción del ambiente. Para el desarrollo de estos estudios se debe tener en cuenta los siguientes puntos.

- Para la muestra de la reconstrucción 3D se van a utilizar dos imágenes que se llamarán imagen 1 y 2 respectivamente. Una vez que la cámara toma la imagen 1, sufre un desplazamiento respecto desde ese punto tomando la imagen 2. Las dos imágenes contienen información del mismo objeto.

- Estas imágenes son comparadas y definidas para que el algoritmo pueda trabajar con ellas. De este modo, el algoritmo extrae los puntos reales de las dos imágenes.

- Se representan los puntos reales de las dos imágenes en el espacio.

Se dispone de las dos imágenes tomadas por la cámara mostradas en la figura 4.3. Entre ellas la cámara ha experimentado un desplazamiento muy pequeño (unos 2 cm).



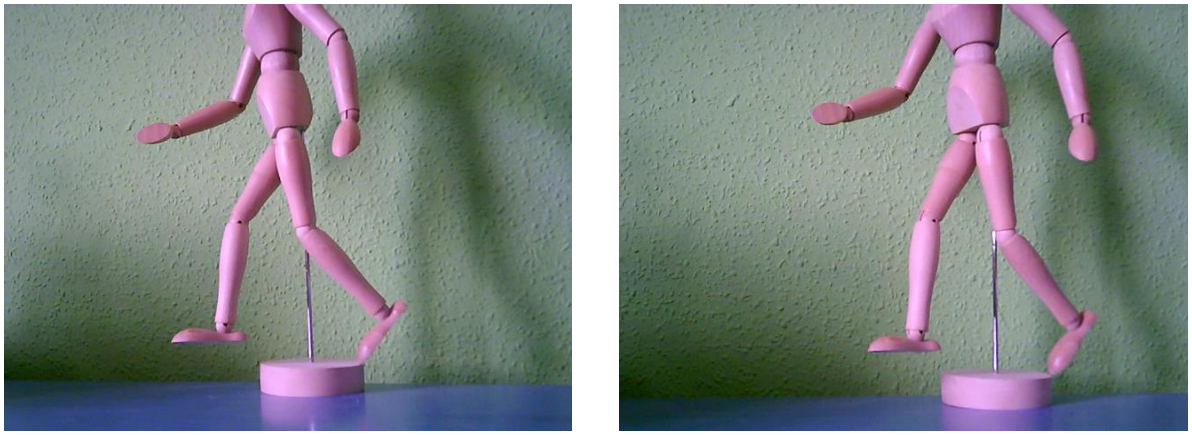
(a) Imagen 1 (b) Imagen 2

Figura 4.3: *En este caso, la imagen 1 y 2 apenas han tenido desplazamiento entre sí, por lo que existen muchos puntos en común.*

A partir de estas dos imágenes se obtiene la representación de los puntos reales en el espacio de la figura 4.4. Se puede apreciar que en la reconstrucción se muestran puntos de partes del muñeco bien definidas como son las piernas y los brazos, aunque también muestra muchos puntos del fondo. En este caso y dado que apenas se considera desplazamiento entre las dos imágenes, la reconstrucción 3D muestra bastante información fiable del objeto.

Figura 4.4: *Reconstrucción 3D de las figuras 4.3 (a) y (b)*

En el siguiente caso, el desplazamiento de la cámara entre las dos imágenes es mayor (de unos 15 cm), por lo que las dos imágenes tienen menos información en común. En la imagen 4.5 se puede apreciar que existe un desplazamiento y una rotación de la cámara, por lo que existen menos puntos en común.



(a) Imagen 1 (b) Imagen 2

Figura 4.5: *En este caso, la imagen 1 y 2 han tenido un desplazamiento considerable, entonces no todos los puntos están contenidos en las dos imágenes.*

La imagen 4.8 muestra la posición de los puntos reales en el espacio. En esta imagen se aprecia que el objeto no aparece tan definido. El desplazamiento y rotación grandes de la cámara hace que los puntos de información coincidentes en ambas imágenes sean muchos menos. En esta reconstrucción siguen apareciendo muchos puntos del fondo que no aportan definición a la imagen.

En los anteriores ejemplos se tenía una distancia a la cámara pequeña, puesto que las imágenes estaban tomadas a un muñeco a 1 metro de la cámara. En la vida real, el UAV no tomará imágenes en esta situaciones. Para poder estudiar el efecto de la reconstrucción 3D las imágenes deben ser aéreas.

En las imágenes de la figura 4.7 se puede apreciar este caso, siendo dos imágenes con bastante profundidad.

Si se consigue la reconstrucción 3D de estas dos imágenes se obtiene la figura 4.8. En este caso se consiguen bastantes puntos 3D representados. En los casos anteriores, cada píxel contenía una cantidad de información muy alta por lo que la coincidencia entre las

Figura 4.6: *Reconstrucción 3D de las figuras 4.5 (a) y (b)*

(a) Imagen 1 (b) Imagen 2

Figura 4.7: *En este caso, la imagen 1 y 2 son aéreas, no dando tanto importancia al detalle como en los casos anteriores, lo que hace que los keypoints tengan más fácil coincidir en las dos imágenes.*

Figura 4.8: *Reconstrucción 3D de las figuras 4.7 (a) y (b)*

dos imágenes era difícil. En este caso, al tener las imágenes más profundidad el nivel de las características que debe almacenar cada píxel es menor, favoreciendo la coincidencia en las dos imágenes. El desplazamiento de las dos imágenes es grande (15 cm) pero a diferencia del segundo caso, esta distancia no afecta.

Por lo tanto, el uso del algoritmo basado en Structure from Motion será válido para aplicaciones aéreas, siendo un método ideal para su utilización en UAVs.

COMENTARIOS

A la vista del estudio de las imágenes y su reconstrucción 3D se pueden hacer algunos comentarios:

- La calidad de los puntos representados varía en función de las condiciones de la cámara. Para distancias cortas a la cámara, si el desplazamiento es mínimo la calidad de la reconstrucción aumenta considerablemente mientras que si el desplazamiento o la rotación son grandes, la calidad de la reconstrucción baja. Con desplazamientos grandes se alteran varias condiciones del objeto como son la rotación, la escala y la iluminación. Si bien el método SIFT-FAST es robusto ante estos cambios, al ser tan grande acaba afectando a la reconstrucción. Por lo tanto, una idea que se puede extraer es que los desplazamientos y las rotaciones deben ser mínimos para una

óptima reconstrucción.

- En el caso que las imágenes sean aéreas, la reconstrucción 3D basada en Structure from Motion y el uso del método SIFT-FAST se presenta como una buena opción, ya que los problemas aplicados a pequeña distancia de la cámara no aparecen consiguiendo una buena calidad de los puntos representados.

- Dentro de la reconstrucción 3D aparecen muchos puntos representados que no aportan información como pueden ser puntos del fondo de la imagen. Estos puntos no han sido filtrados en los pasos previos a la obtención de la reconstrucción 3D de manera óptima, por lo que aparecen en la reconstrucción y restan calidad a ésta. Una de las líneas de mejora del trabajo sería ésta.

Capítulo 5

CONCLUSIONES Y TRABAJOS FUTUROS

El proyecto aborda el problema de la reconstrucción 3D.Como se ha visto, existen diversas alternativas para conseguir representar los puntos reales. De entre todas las posibles se escoge el algoritmo basado en el concepto Structure from Motion, que centra su trabajo en el estudio y tratamiento de imágenes en movimiento. Para conseguirlo se elabora un algoritmo iterativo para la obtención de los puntos reales, basado en la extracción de la información del entorno, su tratamiento y su transformación. El objetivo es la evaluación de las prestaciones de este algoritmo.

5.1. CONTRIBUCIONES DEL PROYECTO

Las principales contribuciones que se han realizado son:

- Se realiza un estudio del arte de las diversas formas de extracción de la información del entorno. Se realiza una explicación de los métodos de extracción de keypoints y descriptores SIFT, SURF, FAST y FREAK, analizando su funcionamiento y viendo sus ventajas e inconvenientes.
- Se realiza un algoritmo de representación de los puntos reales, explicando el funcionamiento de cada etapa del algoritmo. Se presenta un estudio del funcionamiento

de cada elemento participante y su uso dentro del algoritmo.

- Desarrollo del algoritmo en lenguaje C# para la muestra de resultados. Se justifica la elección de los métodos de extracción de keypoints, de su filtrado y su procesado, así como de la representación de resultados prácticos de la reconstrucción 3D.

5.2. CONCLUSIONES SIGNIFICATIVAS

El estudio propone la representación de los puntos reales del entorno mediante el tratamiento de las imágenes. En este sentido el proyecto ofrece los siguientes resultados.

- La extracción de los keypoints tiene mucha importancia debido a que la representación de los puntos reales viene determinada por la obtención de dichos keypoints. Durante el proyecto se han propuesto diversos métodos para la extracción de esta información. Se ha optado por un método conjunto SIFT-FREAK. La obtención de los keypoints mediante el algoritmo SIFT y de los descriptores mediante el método FREAK ofrece una buena definición de las características y también robustez ante cambios en la rotación, la escala y el ruido.

- El filtrado de los keypoints consigue aumentar la calidad de la reconstrucción, por lo que hay que prestarle mucha atención. La elección de métodos de comparación como BruteForce y métodos de exclusión de extremos como el método RANSAC adquieren mucha importancia y tienen que ser bien utilizados para la buena calidad del producto.

- Respecto de la reconstrucción 3D, se aprecian varios puntos reseñables. En primer lugar, el desplazamiento y rotación de la cámara afectan al resultado final. Si se consigue minimizar tanto el desplazamiento como la rotación la calidad de los puntos obtenidos aumenta de forma considerable. La solución a este problema es aumentar el fondo de las imágenes, siendo las imágenes aéreas la mejor opción. Otro punto a destacar es la importancia de un buen filtrado en las etapas anteriores. Si el

filtrado no es bueno aparecen muchos puntos de falsa información que distorsionan el resultado final.

- Por último y a modo de explicación, la elección del método Structure from Motion resta bastante definición a la calidad final de los puntos dado su carácter dinámico. Esta hipótesis es asumida desde un principio, siendo la calidad inferior a una reconstrucción con cámaras estéreo.

5.3. PERSEPECTIVAS Y TRABAJO FUTURO

Durante la realización de este proyecto han ido surgiendo ideas de mejoras que no han sido aplicadas por falta de medios o tiempo de investigación. Algunas de ellas son expuestas en este aparatado en forma de posibles mejoras para la optimización del programa.

- El método elegido para la obtención de los keypoints ha sido el método SIFT. Unos de los puntos en contra que tiene este método es su tiempo de computación, elevado si se compara con otros métodos como SURF o FAST. Existen en el actualidad mejoras del método SIFT en tiempo de computación que sería interesante estudiar para su aplicación como se recogen en el artículo [13] de la bibliografía.

- Además de los métodos de extracción de keypoints y descriptores explicados en este proyecto, existen más que no han sido incluidos por falta de tiempo. Algunos ejemplos son los métodos ORB, una optimización del método FAST, o el método BRIEF.

- Dado que en la reconstrucción 3D que hemos presentado se filtran varios puntos que no deseamos, una de las posibles líneas de investigación para la optimización de este método es la mejora del filtrado previo. Para ello, se pueden buscar otras alternativas a los algoritmos de comparación o de discriminación de extremos.

APÉNDICES

APÉNDICE A

CALIBRACIÓN DE LA CÁMARA

El objetivo de este apéndice es la obtención de la matriz de calibración de la cámara. Para ello, se va a desarrollar el concepto de matriz de calibración y se va a explicar el proceso para su obtención.

A.1. FUNDAMENTOS TEÓRICOS

La calibración de una cámara se puede definir como el proceso de estimación de sus parámetros.

Cuando se busca hacer una representación 3D a través de puntos en 2D, se define la siguiente ecuación:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A[R \mid t] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Donde el vector $[u, v, 1]^T$ representan la posición de los puntos en 2D, $[x, y, z, 1]^T$ representan las coordenadas de los puntos en 3D y $A[R \mid t]$ representan las características de la cámara. Estas características se pueden clasificar como extrínsecos, las variables $[R \mid t]$ correspondientes a la rotación y traslación de la cámara, o intrínsecos, la variable A.

Los valores extrínsecos como son la traslación y la rotación se van a desarrollar a lo largo del informe. En este apartado se van a estudiar los valores intrínsecos, aquellos valores que dependen exclusivamente de la construcción de la cámara. Se puede definir A como una matriz 3x3 con la siguiente estructura:

$$A = \begin{bmatrix} Fc_x & \gamma & cc_x \\ 0 & Fc_y & cc_y \\ 0 & 0 & 1 \end{bmatrix}$$

Donde Fc denota la distancia focal de la cámara, cc denota la distancia al centro óptico, que idealmente se podría poner en el centro de la cámara, y γ es el coeficiente de inclinación, que generalmente es 0.

Todos los datos pueden ser obtenidos mediante algoritmos de computación. En el siguiente punto se estudiará uno de esos algoritmos para la obtención de los valores intrínsecos de la cámara.

A.2. OBTENCIÓN DE LOS VALORES INTRÍNSECOS DE LA CÁMARA

Para la obtención de estos valores se puede recurrir a muchos algoritmos por computadora. En concreto se va a utilizar un algoritmo para Matlab llamado *TOOLBOX_calib*.

Para el funcionamiento del algoritmo, se van a tomar varias fotos en diferentes posiciones a una cuadrícula en forma de tablero de ajedrez y se va a proceder a detectar sus esquinas. Con estos datos el algoritmo será capaz de detectar los valores intrínsecos de la cámara.

DATOS DE ENTRADA EN EL ALGORITMO

Para el análisis de la cámara es necesario introducir varias imágenes de una cuadrícula en forma de tablero de ajedrez, tal como se muestran en la figura A.1. Es importante saber las medidas de los lados de cada cuadrado de la cuadrícula para obtener una buena calibración.

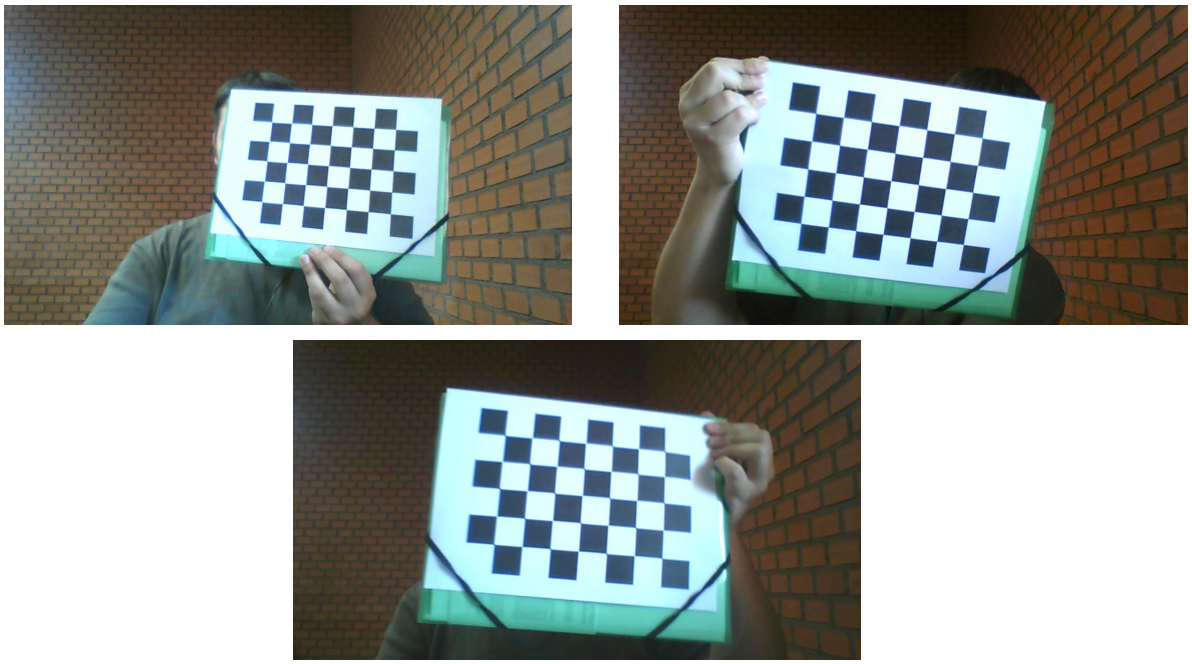


Figura A.1: *Para poder calibrar la cámara hay que hacer algunas fotos a una cuadrícula para extraer sus características.*

En total se han tomado 20 imágenes, considerando una buena base de datos para la calibración. Sin embargo, se pueden escoger el número de imágenes que se quiera, sabiendo que a mayor número mejor será la calidad.

Dichas imágenes se han de nombrar "$ImageXX.jpg$" donde XX indica el número del archivo. De esta forma el programa puede acceder de manera rápida a todos ellos. Se puede usar cualquier formato de imagen, en este caso se ha escogido un formato *.jpg*.

ENTORNO DE LA HERRAMIENTA EN MATLAB

Esta herramienta trabaja en el entorno de Matlab, siendo una de las más precisas de las existentes hoy en día. Se puede descargar de su sitio web [1].

Hay que descomprimir el archivo y dejarlo en un lugar conocido. Se recomienda poner las imágenes tomadas anteriormente en este directorio por facilidad.

Se accede a la herramienta mediante Matlab y se ejecuta con los siguientes comandos:

```
>> calib
```

Aparecerá la pantalla de la figura A.2, desde donde se puede gestionar todas las acciones del algoritmo.

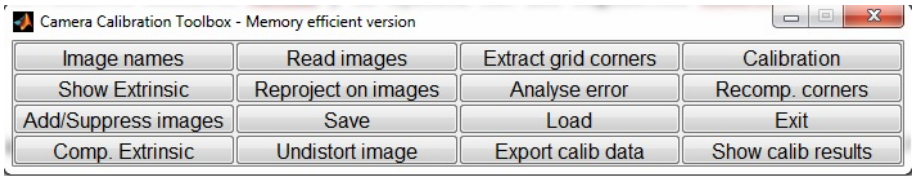


Figura A.2: *Menú de la calibración de la herramienta de Matlab*

El siguiente paso es leer y obtener las imágenes. Para ello se pulsa el botón Images names. En la ventana de Matlab hay que poner la raíz del nombre de las imágenes (en este caso es Image) y seleccionar el formato (en este caso *.jpg*). De vuelta en el menú se pulsa el botón Read Images para que la herramienta las registre.

A continuación se procede al análisis de las imágenes. Para ello hay que pulsar Extract Grid Corners. En este apartado se van a marcar las esquinas de todas las imágenes para su calibración. El marcado se realiza en las 20 imágenes tal y como se aprecia en la figura A.3.

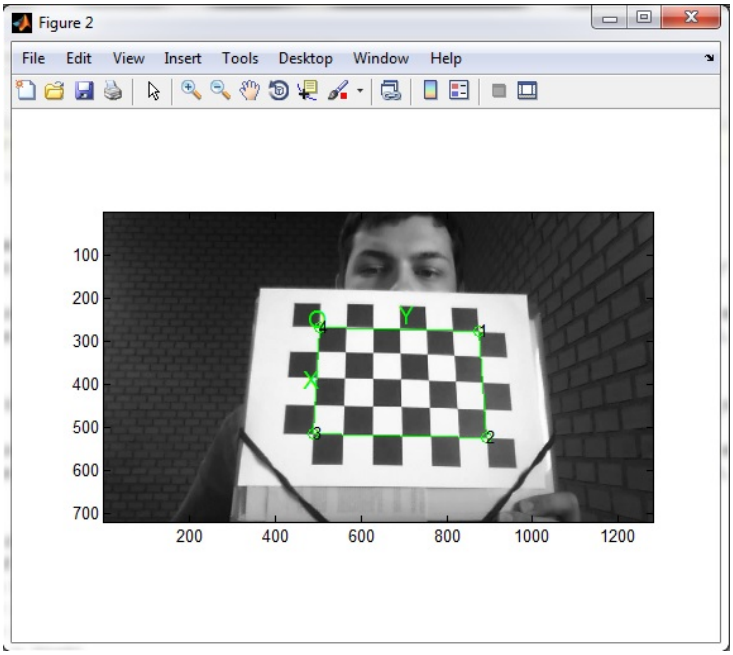


Figura A.3: *Marcación de las esquinas para la calibración*

Una vez marcada la primera imagen se especifica en la ventana de Matlab las dimensiones de los cuadros de la cuadrícula. A partir de aquí se procede a la calibración de las esquinas de las 20 imágenes. Al finalizar hay que volver a la pantalla de menú y elegir la opción Calibrate. En pantalla se muestran los valores de la distancia focal (Fc) y del centro óptico (cc) que se utilizan para definir la matriz decalibración de la cámara.

Por último, si se pulsa en el botón Show Extrinsic se mostrará la posición de la cámara y las diferentes imágenes en el plano, como se muestra en las figuras A.5 y A.4

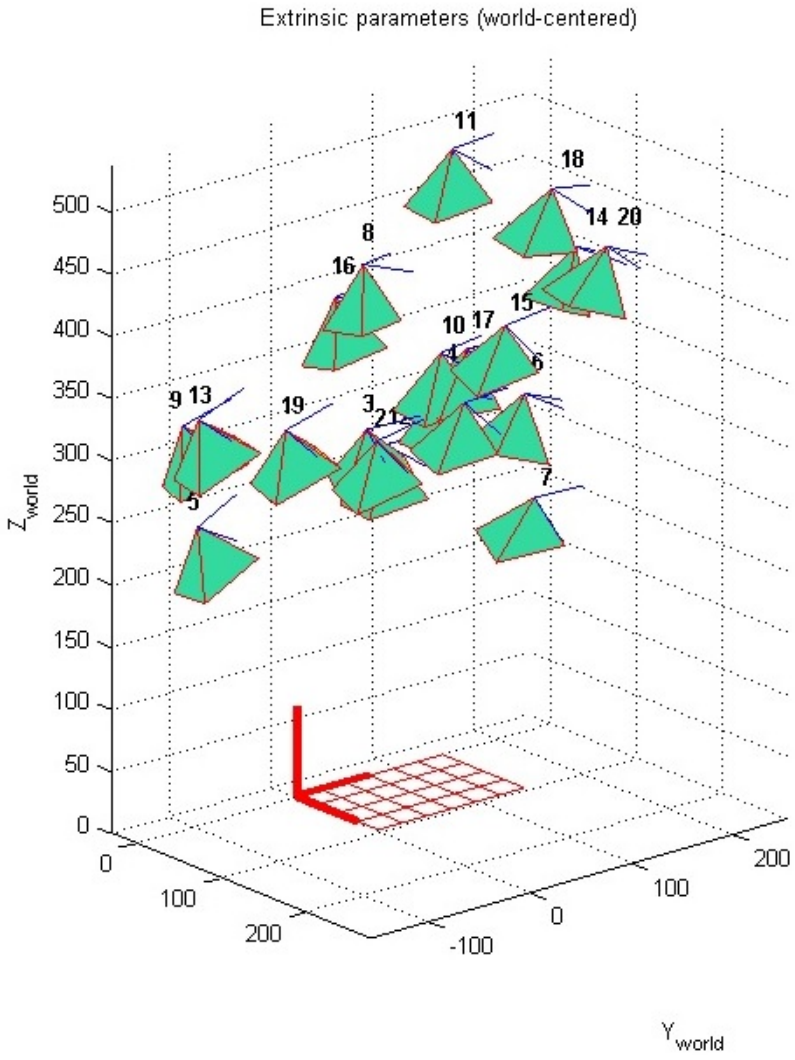


Figura A.4: *Posición de las imágenes centrados en el mundo.*

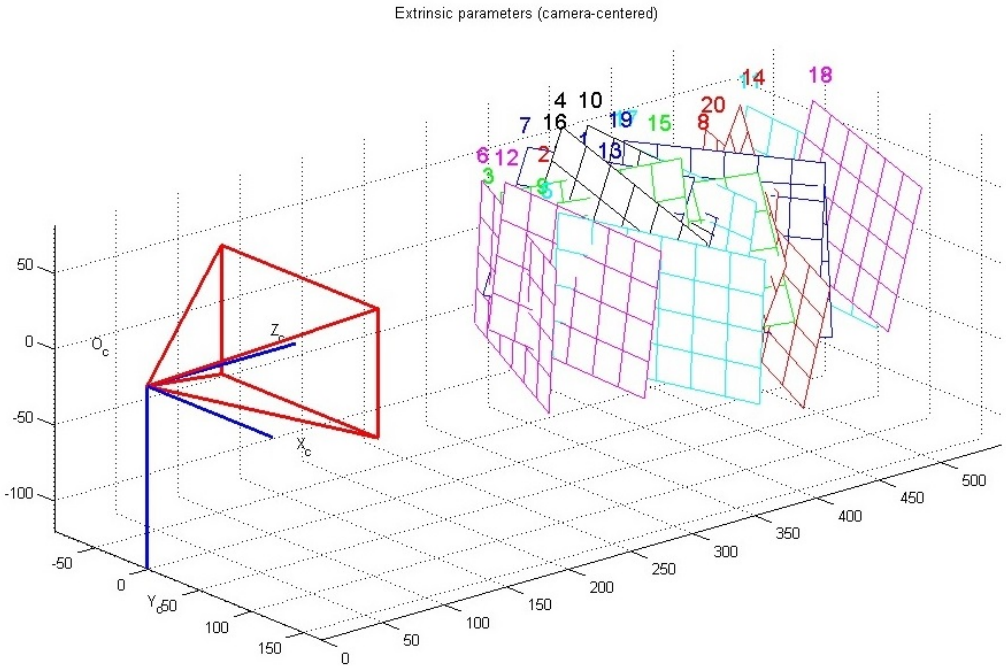


Figura A.5: *Posición de las imágenes referenciados centrados a la cámara.*

APÉNDICE B

PRESUPUESTO DEL PROYECTO

En este apéndice se presentan justificados los costes globales de la realización de este Trabajo de Fin de Grado. Tales costes, imputables a gastos de personal y de material, se pueden deducir de las Tablas B.1 y B.2.

Tabla B.1: *Fases del Proyecto*

Fase 1	*Documentación*	150 *horas*
Fase 2	*Desarrollo del algoritmo*	300 *horas*
Fase 4	*Redacción de la memoria*	150 *horas*

En la Tabla B.1 se muestran las fases del proyecto y el tiempo aproximado para cada una de ellas. En la figura B.1 se puede ver una aproximación de los plazos de desarrollo de cada una de las fases. Así pues, se desprende que el tiempo total dedicado para la realización del trabajo ha sido de 600 horas, de las cuales aproximadamente un 15 % han sido compartidas con el tutor del proyecto, por lo que el total asciende a 690 horas. El mercado actual está liberado de la imposición de precios fijados, por lo que se hace una estimación del honorario del ingeniero. Así, un precio de 25€/hora brutos se considera una cifra razonable. El importe total del coste en personal de este proyecto asciende a 17250 €.

Tabla B.2: *Costes de material*

Ordenador portátil de gama media	600 €
Cámara USB	24 €
Licencia software Visual Studio 2013 Professional	1089 €
Librerías emguCV de tratamiento de imágenes	Gratuito
Latex - Editor de texto	Gratuito

En la Tabla B.2 se recogen los costes de material desglosados en el equipo hardware utilizado y el software. Ascienden, pues, a un total de 1713 €.

Tabla B.3: *Presupuesto*

Concepto	**Importe**
Costes personal	17250 €
Costes material	1713 €
TOTAL	111.893,6 €

A partir de estos datos, el presupuesto total es el mostrado en la Tabla B.3, que asciende a un total de 18963 €.

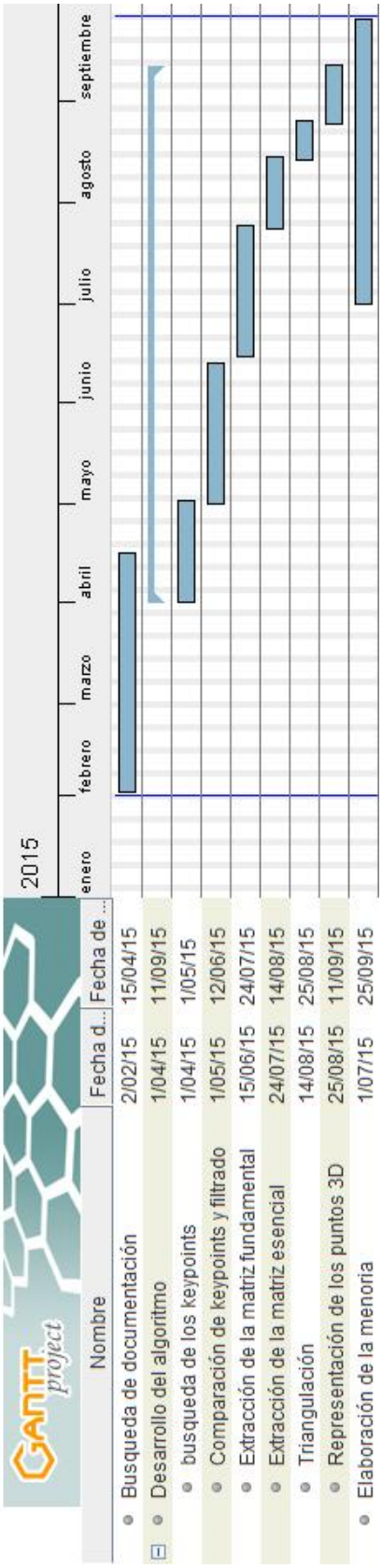


Figura B.1: *Diagrama de Gantt con las etapas desarrolladas a lo largo del proyecto y su estimación en tiempo.*

Bibliografía

[1] *http* : *//www.vision.caltech.edu/bouguetj/calib_doc/*. Última revisión: 25/09/2015.

[2] Dictionary of military and associated terms. *Department of Defense U.S.*, 31 /10/2009.

[3] *https* : *//en.wikipedia.org/wiki/scale* − *invariant_feature_transform*. Última revisión 25/09/2015.

[4] *https* : *//en.wikipedia.org/wiki/singular_value_decomposition*. Última revisión: 25/09/2015.

[5] H. Aanæs and R. Larsen. *Methods for structure from motion*. PhD thesis, Technical University of DenmarkDanmarks Tekniske Universitet, Department of Informatics and Mathematical ModelingInstitut for Informatik og Matematisk Modellering, 2003.

[6] A. d. l. E. Abdulla Al-kaff and J. M. Armingol. Sift and surf performance evaluation and the effect of freak descriptor in the context of visual odometry for unmanned aerial vehicles.

[7] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517. Ieee, 2012.

[8] D. L. Baggio. *Mastering OpenCV with practical computer vision projects*, volume Chapter 4: Exploring Structure from Motion Using OpenCV. Packt Publishing Ltd, 2012.

[9] A. Barrientos, J. del Cerro, P. Gutiérrez, R. San Martín, A. Martínez, and C. Rossi. Vehículos aéreos no tripulados para uso civil. tecnología y aplicaciones. *Universidad politécnica de Madrid, Madrid*, 2007.

[10] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.

[11] R. R. Delgado. Los drones y sus aplicaciones a la ingeniería civil. Capítulo 14. Aplicaciones al mantenimiento de líneas eléctricas:175 – 184, 2015.

[12] M. I. García. Algoritmos de visión para la estimación robusta de pose 3d. pages 67–82, Octubre 2010.

[13] M. Grabner, H. Grabner, and H. Bischof. Fast approximated sift. In *Computer Vision–ACCV 2006*, pages 918–927. Springer, 2006.

[14] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*, volume Chapter 9: Epipolar Geometry and the Fundamental Matrix. Cambridge university press, 2003.

[15] R. I. Hartley and P. Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.

[16] Y. Huang, S. J. Thomson, W. C. Hoffmann, Y. Lan, and B. K. Fritz. Development and prospect of unmanned aerial vehicle technologies for agricultural production management. *International Journal of Agricultural and Biological Engineering*, 6(3):1–10, 2013.

[17] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[18] J. M. Peña-Barragán, F. López-Granados, L. García-Torres, M. Jurado-Expósito, M. S. De La Orden, and A. García-Ferrer. Discriminating cropping systems and agro-environmental measures by remote sensing. *Agronomy for Sustainable Development*, 28(2):355–362, 2008.

[19] C. C. Rejado. Los drones y sus aplicaciones a la ingeniería civil. Capítulo 1. Origen y desarrollo de los Sistemas de Aeronaves Pilotadas por Control Remoto:15 – 32, 2015.

[20] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1508–1515. IEEE, 2005.

[21] Z. Ting. Emgucv image process: Estimating projective relations in images, Septiembre 2015. $http://yy-programer.blogspot.com.es/2013/07/emgucv-image-process-estimating\_24.html$.

[22] D. G. Viswanathan. Features from accelerated segment test (fast), 2009.